Notes on Channel Routing with Knock-Knees

Maria Artishchev-Zapolotsky
zmaria@cs.technion.ac.il
Computer Science Department
Technion, Israel Institute of Technology
Haifa, Israel 32000

Abstract

In VLSI layout of interconnection networks, routing two-point nets in some restricted area is one of the central operations. It aims usually to minimize the layout area. Here, we consider connecting (with knock-knees) sets of N inputs and N outputs on the opposite sides of a rectangular channel, where the output order is a given permutation of the order of corresponding inputs. Such channels are used in many types of VLSI layouts, among them interconnection networks. Here we complete the algorithm of Muthukrishnan et al. and correct its analysis, and adjust the algorithm of Pinter for the same problem conditions.

Keywords: VLSI layout, channel routing, area optimization.

1 Introduction

1.1 Problem Background

The construction of efficient VLSI layouts for interconnection networks is important, since it improves performance of the resulting parallel architecture. Routing two-point nets in some restricted area is one of the central operations in this problem class. A vast amount of work has been done on routing in a channel (see e.g. [1, 2, 3, 4]). The main aim of these works is minimization of the layout area.

In this paper, we consider the channel routing problem (as defined in [3]) where the terminals are placed just on two opposite sides of the channel boundary. This definition also allows *knock-knees* (when two wires share a grid point and bend at this point).

It is a typical situation when a set of the net terminals is placed on a boundary of some area, and the wiring has to be performed inside this area. Sometimes the area is not rectangular; and the terminals may not be concentrated only on the opposite sides. However, there are heuristic solutions of this problem, allowing to divide the wiring area into rectangular channels. Then, the routing may be done separately for each such channel, that is usually much easier than dealing with the whole (original) problem.

The problem of rectangular channel routing is studied in [3, Theorem 1] (as a part of interconnection network layout). There is also an earlier solution for this problem, but with knock-knees forbidden, in [4]. Here, we focus on these two methods and refine and/or modify them, as applied to the setting of [3].

1.2 Our Results

The algorithm described in Theorem 1 in [3] consists of 2 parts. The first part of the algorithm is detailed, but analyzed not exactly: the number of columns used by the resulting layout is overestimated. The second part is erroneously claimed be trivial and is not described; the claimed bound on the number of columns and the number of wire bends in this part are wrong. Thus the algorithm is not complete, and the bound claimed in the theorem is not valid.

Our first result is completing the algorithm from [3, Theorem 1] and correcting its analysis. The second one is construction of an algorithm similar to that of Pinter [4] for the setting of [3]. That is:

- We refine the analysis of the first part of the algorithm from [3].
- We complete the second part of the algorithm.
- We prove *tight* bounds for the number of columns occupied by the layout and for the number of bends per wire in it.

- We adjust the algorithm of Pinter [4] to the conditions of this problem (allowing knock-knees), and show that it is simpler and does not use additional columns as the algorithm from [3, Theorem 1] does.

2 Problem Definition

There is an integer grid, where rows are numbered from the top down, and columns — from the left to right. There are n inputs $I_1, I_2, \ldots, I_n$ lying in the same column and each I_i in i's row. The same for the outputs $O_1, O_2, \ldots, O_n$.

Let f denote a permutation $f : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$. The goal is to produce a layout of this permutation: to connect each I_i to $O_{f(i)}$ via a set of edge-disjoint paths in the grid (see Figure 1).

3 Layout Algorithm

3.1 Auxiliary Graph

We describe here the construction from [3]. For a given permutation $f : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$, a graph $G(f)$ is built with vertices $V = \{1, \ldots, n+1\}$ and edges $E = \{(i, f(i)+1) \mid 1 \leq i \leq n\}$.

In $G(f)$, a path P is *maximal decreasing* (*maximal increasing*) if the sequence of vertex numbers along P is decreasing (increasing) and no extension of P has this property. Note that such paths in $G(f)$ are defined in a unique way. We are interested in the total number of maximal decreasing and maximal increasing paths, which we denote by $m(G(f))$, or simply $m(G)$; trivially, $m(G) \leq n$. This number will designate a width of the channel layout. For a given path P_0, the total number of maximal decreasing and maximal increasing paths in P_0 is denoted similarly — $m(P_0)$.

For convenience, let g denote the mapping defined by $G(f)$, i.e. $\forall\ 1 \leq i \leq n \quad g(i) = f(i) + 1$. And h will denote the mapping from image of g to image of f, in other words $h(g(i)) = f(i)$. Clearly, here $h(j) = j - 1$ for each $1 < j \leq n+1$.

In $G(f)$, every vertex except $n+1$ has outdegree one and every vertex except 1 has indegree one. For $n+1$ the outdegree is zero, and for 1 the indegree is zero. Clearly, if $G(f)$ has a Hamiltonian path, then that path goes from 1 to $n+1$. Otherwise, $G(f)$ consists of a path from 1 to $n+1$ and a set of cycles.

3.2 Layout Construction

The layout uses rows $1, \ldots, n+1$, where row i contains input node I_i, and output node O_i. The layout consists of two parts:

1. Main-stage: wiring mapping g using $m(G)$ columns. This wiring, actually, connects each input I_i (placed in line i) to an intermediate point M_j placed in line j (see Figure 1), where $(i, j) \in G(f)$, i.e. $j = g(i)$.
2. Fix-stage: wiring each output of the first stage – M_i – to the global output $O_{h(i)}$. This stage, actually, restores the original permutation by performing the mapping h on the intermediate outputs.

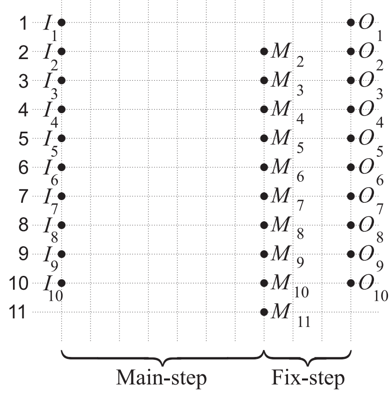


Figure 1: The layout scheme.

For main-stage, 2 cases are distinguished:

1. If $G(f)$ has (i.e. is) a Hamiltonian path, the column assignment to connections is as follows. Consider the maximal increasing path ending at $n+1$, let it start at some j_1. For each edge $(i, f(i)+1)$ on this path, we assign column 1 to the connection from I_i, to $M_{g(i)}$. Next we find the maximal decreasing path ending at j_1, and starting at some j_2. We assign column 2 to the connections corresponding to edges on this path, and so on.
2. If $G(f)$ does not have a Hamiltonian path, then $G(f)$ is transformed into new graph $G'(f)$ that does have a Hamiltonian path. And then the mapping g is defined by the new graph $G'(f)$ and the mapping h is changed accordingly. The transformation of $G(f)$ to $G'(f)$ is made in the following way:
   (a) Let $G'(f) \leftarrow G(f)$.
   (b) Let P denote the path in $G'(f)$ from 1 to $n+1$.
   (c) While there are vertices in $G'(f)$ that are not on P, do:
      i. Choose the maximal $j \notin P$.
      ii. Replace the edges (a, j) and $(b, j+1)$ by the new ones – $(a, j+1)$ and (b, j).

   This new graph G' is Hamiltonian. Now, the main-stage of wiring of f can be completed in $m(G')$ (= the total number of maximal decreasing and maximal increasing paths in $G'(f)$) columns, using the algorithm for Hamiltonian path layout. The authors of [3] claim that $m(G') \leq m(G) + 1$. We refine this statement proving that $m(G') \leq m(G)$.

4 Corrections of Theorem 1 from [3]

Theorem 1 (from [3]) *There is an $O(n)$ time algorithm to produce a layout of f using $n+1$ rows and at most $m(G)+2$ columns. Each connection has at most four bends.*

We show that the above construction (from the theorem proof) may use more columns $(m(G)+3)$ and there are wires with 6 bends, and we prove that these are also the upper bounds. Our proof is based on the original proof of the theorem with some corrections:

1. The theorem proof claims that the main-stage of the layout uses $m(G)+1$ columns in the worst case. We prove in Lemma 3 that $m(G)$ is an upper bound.
2. The fix-stage is not described and is supposed, by the theorem proof, to consume only a single column. In Lemma 4, we prove a lower bound 3 in the worst case, for it. We suggest a solution to the fix-stage, and prove for it a bound of 3 columns. Thus, the overall width of the layout increases to $m(G)+3$ in the worst case.

   As well, we prove that $m(G)+3$ is also an upper bound (Lemma 4).
3. In Lemma 5, we show that some wires make 4 bends in the fix-stage, thus the lower bound of the overall number of bends for such wires is 6.

   We prove that 6 is an upper bound too.

The rest of this section describes corrections to Theorem 1.

Theorem 2 *The algorithm produces a layout of f using $n+1$ rows and at most $m(G)+3$ columns. Each connection has at most six bends. These bounds are tight.*

Proof: Follows directly from Lemmas 3, 4, 5. ■

Lemma 3 *Let $G(f)$ be the auxiliary graph (as constructed in subsection 3.1) that does not contain Hamiltonian path. Let $G'(f)$ be the transformation of $G(f)$ (as described in subsection 3.2, main-stage, case 2). Then $m(G') \leq m(G)$.*

Proof: Some auxiliary definitions:

- Let $u \rightsquigarrow v$ denote a path from u to v in G.
- For any path $u \rightsquigarrow v$, let $m(u \rightsquigarrow v)$ denote the total number of maximal decreasing and maximal increasing paths in $u \rightsquigarrow v$.
- Vertex j is a *peak*, if some maximal increasing/decreasing path starts/ends at j.

The transformation from G to G' is made iteratively by integrating each separate cycle into the path from 1 to $n+1$. Let us name the process of a single cycle integration into the path by "step". We show that in each such step the total number of maximal decreasing and maximal increasing paths does not grow, thus proving the Lemma.

Consider a single step:

- Let P denote the path $1 \rightsquigarrow (n+1)$ at the beginning of the step (before the cycle integration process started).
- Let C denote the cycle to be integrated. Reminder: at each step, the maximum $j \in V \setminus P$ is chosen. So, C is exactly the cycle containing j.
- Let P' denote the path $1 \rightsquigarrow (n+1)$ at the end of the step (after C is integrated).

Since j is the maximum among all the vertices that are not on P, it is also maximum in C. Therefore j is a peak in C. Let $j \rightsquigarrow x$ be a maximal decreasing path in C, and let $y \rightsquigarrow j$ be a maximal increasing path in C (it is possible that $x = y$), then $m(C) = m(j \rightsquigarrow j) = 1(\text{for } j \rightsquigarrow x) + m(x \rightsquigarrow y) + 1(\text{for } y \rightsquigarrow j)$.

Note that $j+1 \in P$ (since j is maximum among those who is not on P). Let $(a, j) \in C$, $(b, j+1) \in P$. Then the integration of C into P is made by replacing (a, j) and $(b, j+1)$ with (b, j) and $(a, j+1)$.

Since all the numbers are unique, the path $y \rightsquigarrow (j+1)$ is still increasing, and the relation between b, j is the same as between $b, j+1$. Therefore, $m(1 \rightsquigarrow (j+1))(\text{in } P) = m(1 \rightsquigarrow j)(\text{in } P')$ and $m(j \rightsquigarrow j)(\text{in } C) = m(j \rightsquigarrow (j+1))(\text{in } P')$.

Let us consider 3 cases:

1. If $j+1$ is a peak, then $m(P) = m(1 \rightsquigarrow (j+1)) + m((j+1) \rightsquigarrow (n+1))$. Then $m(P') \leq m(1 \rightsquigarrow j) + m(j \rightsquigarrow (j+1)) + m((j+1) \rightsquigarrow (n+1)) = m(1 \rightsquigarrow (j+1)) + m(j \rightsquigarrow j) + m((j+1) \rightsquigarrow (n+1)) = m(P) + m(C)$. Thus, the total number of maximal decreasing and maximal increasing paths in the graph does not grow.
2. If $j+1$ is inside of a maximal increasing path $u \rightsquigarrow v$, then after the integration, $u \rightsquigarrow j$ is still increasing and $y \rightsquigarrow (j+1) \rightsquigarrow v$ is increasing too. So, $m(P') = m(1 \rightsquigarrow u) + 1(\text{for } u \rightsquigarrow j) + m(j \rightsquigarrow y) + 1(\text{for } y \rightsquigarrow v) + m(v \rightsquigarrow (n+1)) = (m(1 \rightsquigarrow u) + 1(\text{for } u \rightsquigarrow v) + m(v \rightsquigarrow (n+1))) + (m(j \rightsquigarrow y) + 1(\text{for } y \rightsquigarrow j)) = m(P) + m(C)$.
3. If $j+1$ is inside of a maximal decreasing path, the proof is symmetric.

■

Lemma 4 *The fix-stage may use 3 columns. And it is an upper bound.*

Proof: In the fix-stage, in the case of $G(f)$ being Hamiltonian, the mapping h is very simple: $h(j) = j-1$. Therefore, its wiring is just shifting each wire a single grid segment upward. It is performed using just a single column.

When $G(f)$ is not Hamiltonian and has to be transformed into $G'(f)$, we distinguish 2 cases:

1. Among all the vertices that are not on P in $G(f)$ there are no two successive ones; i.e. for each $j \notin P$ it holds that $j-1, j+1 \in P$. In this case, if we replace an edge pair (a, j) and $(b, j+1)$ by the new edge pair $(a, j+1)$ and (b, j), we know for sure that neither of the new edges will be replaced again. Then, for each j such that $(a, j) \in G(f)$ and $(a, j) \in G'(f)$ (the edge does not change from G to G'), still holds $h(j) = j-1$. The rest are divided into independent pairs $< j, j+1 >$ such that $(a, j) \in G(f)$ implies $(a, j+1) \in G'(f)$, and $(b, j+1) \in G(f)$ implies $(b, j) \in G'(f)$. For them $h(j) = j, h(j+1) = j-1$. Since these pairs do not overlap, we can still wire them in a single column — for example, see Figure 2.
2. The second case (the problematic one omitted in [3]) contains groups of successive vertices $\{j, j-1, j-2, \ldots, j-k\}$ such that for each $0 \leq i \leq k \;\; j-i \notin P$. In this case, incorporation of each $j-i$ into P affects the origin of the edge $(b, j+1) \in G(f)$.

   More precisely, if $(b, j+1), (a_0, j), (a_1, j-1), (a_2, j-2), \ldots, (a_{k-1}, j-(k-1)), (a_k, j-k) \in G(f)$, and only $j+1$ is on P, then in $G'(f)$ we get the edges: $(a_0, j+1), (a_1, j), (a_2, j-1), (a_3, j-2), \ldots, (a_k, j-(k-1)), (b, j-k)$. Now, let's consider the mapping h for this group: for each $0 \leq i \leq k \;\; h(j-i+1) = f(a_i) = j-i-1$. In other words, $\forall \; j-k+1 \leq i \leq j+1 \;\; h(i) = i-2$. However, $h(j-k) = f(b) = j$.

   Thus, in the group described above there are $k+1$ neighbor wires whose outputs has to be shifted 2 grid segments upward and a wire with an output to be shifted k grid segments downward in the

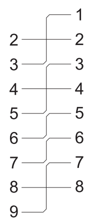


Figure 2: The fix-stage (main-stage is omitted) for the case where the pairs of type $< j, j{+}1 >$ are $< 8, 9 >$, $< 4, 5 >$ and $< 2, 3 >$.

fix-stage. Clearly, the latter wire will occupy k vertical grid segments in one of the columns assigned for the fix-stage wiring. Therefore, all the rest of the wires in the group cannot use this column. Moreover, they cannot use all the same column, because each one needs 2 vertical grid segments. Hence, the number of the columns for the fix-stage wiring is minimum 3, in the worst case (for example, see Figure 3).

Now, let's show that 3 is enough for the fix-stage for any permutation. As we have shown above, all the connections in the permutation can be partitioned into subsets, where each subset contains one of the following:

- A single wire, for which $g(a) = f(a) + 1 = j$ and $h(j) = j - 1$.
- A pair of wires, where $g(b) = j$ and $g(a) = j + 1$. Then $h(j) = j, h(j + 1) = j - 1$.
- A group of wires, where $\forall\ j - k + 1 \le i \le j + 1\ \ h(i) = i - 2$ and $h(j - k) = f(b) = j$.

Neither two of the above subsets overlap, because every subset contains successive numbers only (the inputs for the fix-stage are meant). Therefore, the assignment of the columns for wiring every subset is completely independent. We have shown already that subsets of the first two types (singles and pairs) can be wired using a single column only. For wiring a third type subset, let's assign the columns to the wires in the following way:

- A wire $j - k \to j$ will use the middle column.
- All the wires $(j - k + 2) \to (j - k)$, $(j - k + 4) \to (j - k + 2)$, $(j - k + 6) \to (j - k + 4)$, $\ldots$ will use the rightmost column.
- All the rest will use the leftmost column.

For example, see the Figure 3, where $j = 10$ and $k = 4$. So, $h(6) = 10$, $h(7) = 5$, $h(8) = 6$, $h(9) = 7$, $h(10) = 8$, $h(11) = 9$.

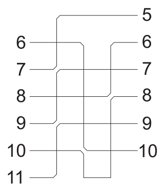


Figure 3: The fix-stage for the case where $j = 10$ and $k = 4$.

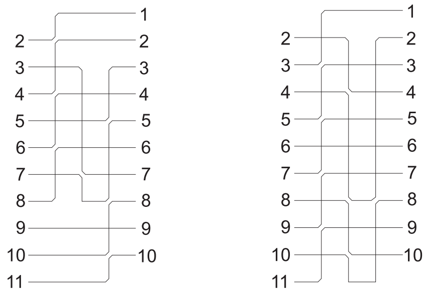
Thus, in the second case, the fix-stage of the layout uses up to 3 columns, and it is also the lower bound. ■

Lemma 5 *Some wires in the layout from [3] (with our addition) may have six bends. And it is also an upper bound.*

Proof: Most of the wires have only 4 bends: 2 in main-stage and 2 in fix-stage. However, the Figure 3 illustrates the case, where some wire in the fix-stage makes 4 bends (a wire $10 \to 8$). Since the wire makes also 2 bends in the main-stage, the total number is 6. Thus, the lower bound for this algorithm (in the worst case) is 6 bends in a single wire.

This happens, actually, for every wire subset of type "group" (see the proof of Lemma 4) where the parameter k is even: in this case, all wires whose endpoints have an even difference with j use the rightmost column (out of 3), and among them also the wire $(j-2) \to j$. But the middle part of the row j is already occupied by the wire $(j-k) \to j$, therefore there is a conflict between these wires. This conflict is resolved by using a grid segment from a lower row (see Figure 3).

This "lower row" is also a subtle point: in the Figure 3, the row $j+1$ is used. However, if $n > j$, then the row $j+1$ is occupied by a wire from the next subset. Then, if the current group-subset is the last of this type (i.e. for each $i > j$, the wire $i \to h(i)$ belongs to single- or pair-subset), the "lower row" is still $j+1$ (Figure 4(a)). Otherwise, the upper row of the next group-subset is used (Figure 4(b)).



(a) (b)

Figure 4: Choosing the "lower row": (a) the group-subset ($j = 7, k = 4$) is the last of this type, (b) the group-subset ($j = 4, k = 2$) is not last, and its "lower row" is the upper row of the next group-subset ($j = 10, k = 2$).

Hence, the only wires that may have more than 4 bends are the wires from group-subsets, but according to the above scheme they can be always completed with 6 bends. So, 6 is also an upper bound. ∎

5 Pinter's Algorithm

The technique of Pinter [4], developed for a knock-knee free channel routing, can also be used, with some modifications, for the problem with knock-knees allowed.

For a given permutation $f : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$, a graph $G^*(f)$ is built with vertices $V = \{1, \ldots, n\}$ and edges $E = \{(i, f(i)) \mid 1 \leq i \leq n\}$.

In $G^*(f)$, every vertex has outdegree one and indegree one. So, each vertex belongs to exactly one cycle. A *start* of each cycle C is the minimal numbered vertex on C. Let $C_1, C_2, \ldots, C_k$ denote all the cycles in $G^*(f)$, sorted in the increasing order of their starts. Clearly, $1 \in C_1$. As well, $m(G^*) = \sum_{i=1}^{k} m(C_i)$.

We wire the cycles independently, each after other, in the defined order starting from C_1. For each cycle C_i, we perform the following operations:

1. Let s denote the start of the cycle. Let $a \leftarrow s$.
2. Assume that the leftmost unused column is number j.
3. Let $a \rightsquigarrow b$ be a maximal increasing path in C_i.
   Assign the column j to the connections defined by the path $a \rightsquigarrow b$.
4. Let $b \rightsquigarrow c$ be a maximal decreasing path in C_i.
   Let $(b, b') \in b \rightsquigarrow c$ be the first edge in the path.
   Then wire the connection (b, b') with a *detour* via the row $(n+1)$: wire it up to the column j, turn to the right and reach the row $(n+1)$. Then pass to the column $(j+1)$ and complete to the row b'.
5. Assign the column $(j+1)$ to the connections defined by the path $b' \rightsquigarrow c$.
6. If $a = c$, then the cycle is completed.
7. Otherwise, let $c \rightsquigarrow d$ be the next maximal increasing path in C_i.
8. Let $(c, c') \in c \rightsquigarrow d$ be the first edge in the path. Then wire the connection (c, c') with a detour via the row s: in the symmetric way as it is done in step 4.
9. Let $a \leftarrow c'$, $b \leftarrow d$. Go to the step 3.

For example, see Figure 5. The permutation f is defined as:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$f(i)$	5	14	13	6	9	16	3	4	7	2	1	10	17	18	11	12	15	8

.

There are two cycles in f: $C_1 = (1, 5, 9, 7, 3, 13, 17, 15, 11)$ and $C_2 = (2, 14, 18, 8, 4, 6, 16, 12, 10)$. Each one consists of four maximal increasing/decreasing paths. For example, the columns 1–4 are occupied by the cycle C_1. The column 1 is used by the connections from the path $(1, 5, 9)$, and the connection $(9, 7)$ makes detour via the row 19.

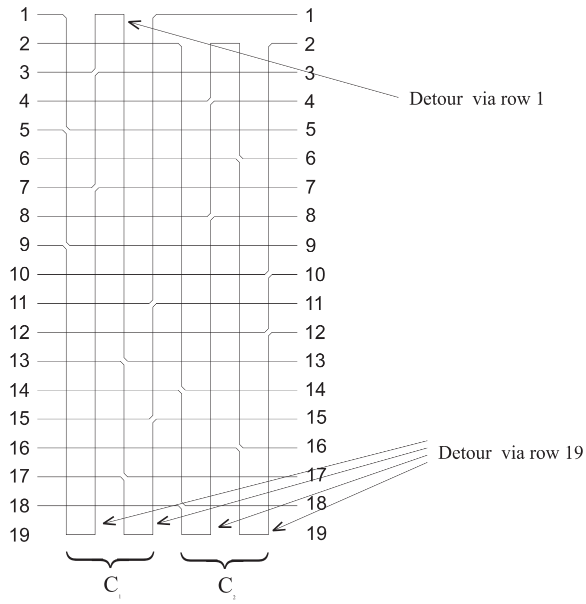


Figure 5: Layout according to Pinter's algorithm of the permutation with two cycles.

Theorem 6 *The algorithm produces a layout of permutation f using $n+1$ rows and $m(G^*)$ columns. Each connection has at most four bends.*

The proof is easy and is omitted.

This algorithm is simpler, as compared with that of Section 4; in particular, there is no need in the second stage. Accordingly, it does not consume any columns in addition to $m(G^*)$.

Acknowledgment

We wish to thank Yefim Dinitz for his numerous suggestions on improving the manuscript.

References

[1] Bonnie Berger, Martin Brady, Donna Brown, and Tom Leighton. Nearly optimal algorithms and bounds for multilayer channel routing. *Journal of ACM*, 42(2):500–542, 1995.

[2] Sandip Das, Susmita Sur-Kolay, and Bhargab B. Bhattacharya. Routing of L-shaped channels, switch-boxes and staircases in manhattan-diagonal model. In *VLSID'98: Proceedings of the Eleventh International Conference on VLSI Design: VLSI for Signal Processing*, pages 65–70. IEEE Computer Society, 1998.

[3] Shan Muthukrishnan, Mike Paterson, Süleyman Cenk Ṣahinalp, and Torsten Suel. Compact grid layouts of multi-level networks. In *31-th ACM Symposium on Theory of Computing (STOC)*, pages 455–463, 1999.

[4] Ron Y. Pinter. On routing two-point nets across a channel. In *ACM IEEE 19th Design Automation Conference Proceedings*, pages 894–902, Los Alamitos, Ca., USA, June 1982. IEEE Computer Society Press.