

# Global Wire Routing in Two-Dimensional Arrays†

(Extended Abstract)

R. M. Karp\*  
F. T. Leighton\*\*  
R. L. Rivest\*\*  
C. D. Thompson\*  
U. Vazirani\*  
V. Vazirani\*

\*Computer Science Division  
573 Evans Hall  
U.C. Berkeley  
California 94720

\*\*Mathematics Department and  
Laboratory for Computer Science  
Massachusetts Institute of Technology  
Cambridge, MA 02139

## Very Brief Abstract

We examine the problem of routing wires on a VLSI chip, where the pins to be connected are arranged in a regular rectangular array. We obtain tight bounds for the worst-case “channel-width” needed to route an  $n \times n$  array, and develop provably good heuristics for the general case. An interesting “rounding algorithm” for obtaining integral approximations to solutions of linear equations is used to show the near-optimality of single-turn routings in the worst-case.

## Problem Definition

We use a classical model wherein the chip area is considered to be divided into a uniform  $n \times n$  array of square *cells*. Each cell contains  $p$  pins (connection points for logic elements). Each instance of our routing problem specifies a collection of *nets* where each net is specified as a set of pins. (Each pin is on at most one net.) Each net is to be connected together by horizontal and vertical wires. Unless stated otherwise, we assume that  $p = 1$  and that each net connects exactly two pins.

A global routing problem instance specifies a *pin placement*, so that the only remaining work is to route the wires between the pins. For this reason, the global routing problem is a special case of (and perhaps easier than) the general placement and routing problem studied in [T79, L80, L81, L82, BL83].

It is common to solve a global routing problem instance  $P$  in two steps:

(1) compute a *global routing*  $R$  specifying for each net the set of cells and cell edges to be traversed by the wiring for that net, and

† This research was supported by funds from Semiconductor Research Corporation grant 1-44247-52055, NSF grant MCS 82-04506, NSF grant ECS 81-10684, NSF grant MCS-80-06938, NSF grant MCS-81-05217, DARPA contract N00014-80-C-0622, Air Force contract OSR-82-0326, a Bantrell Fellowship, and an IBM Graduate Fellowship.

(2) compute a *detailed routing* that specifies for each net the exact position of each wire, which follows the previously computed global routing and satisfies the usual separation constraints between wires, etc.

In this paper we are concerned exclusively with the problem of finding good global routings (which we henceforth call *routings*).

## T-turn Routings

We are particularly concerned with *t-turn routings*, in which the path for each net contains at most  $t$  turns. A one-turn routing will have for each wire either a straight wire segment or an “L”-shaped wire-segment. The number of turns in a global routing is the least number of turns in any detailed routing consistent with the global routing. When horizontal and vertical wires are implemented on distinct layers, then the number of turns required is equal to the number of “vias” or “contact cuts” required to join the straight-line wire segments together. In the general case (e.g. when  $p > 1$ ) we identify the number of “turns” with the number of vias required to implement the wiring pattern, or (equivalently) the sum for each net of the number of cells for which the global routing for that net crosses both a horizontal and a vertical side of the cell.

Notation: We denote the set of global routings for problem instance  $P$  by  $\Gamma(P)$ . The set of  $t$ -turn global routings are denoted by  $\Gamma_t(P)$ .

## Example

Figure 1 presents an example of our global routing problem on a  $4 \times 4$  grid with 8 nets. Figure 2 presents a typical solution to this problem, which happens to be in  $\Gamma_1(P)$ .

		Column			
		1	2	3	4
Row	1	6	6	7	8
	2	5	2	1	3
	3	7	4	1	2
	4	4	8	5	3

Figure 1

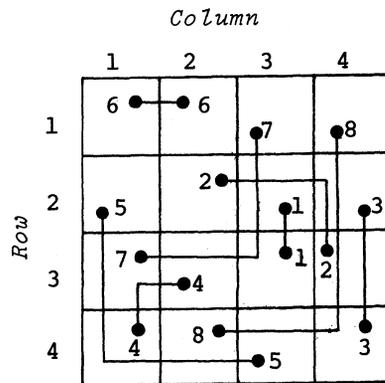


Figure 2

### Channel Widths

Let  $P$  denote an instance of our global routing problem and let  $R$  denote a global routing solving  $P$ .

**Notation:** Let  $w(R)$  denote the maximum number of wires passing from one cell into an adjacent one in the global routing  $R$ .

**Remarks:** Intuitively,  $w(R)$  is the "channel width" which is needed to route the wires of the solution  $R$ , so we call  $w$  the "width" of the solution  $R$ . The one-turn routing  $R$  of Figure 2 has width 3 (there are three wires between cell (2, 4) and cell (3, 4)). Flipping either net 2 or net 8 to its other "L" configuration will reduce the width to 2. The reader can convince himself that no one-turn routing has width one by considering nets 1, 6, and 7.

**Definition:** An optimum global routing  $R$  is one that minimizes  $w(R)$  over all global routings for the given problem instance (i.e. over all  $R \in \Gamma(P)$ ).

**Notation:** (Width of a problem instance  $P$ .) We let  $w(P)$  denote width of an optimal routing  $R$  for  $P$ .

**Notation:** (Width of the best  $t$ -turn routing for  $P$ .) We let  $w_t(P)$  denote the least width of any  $t$ -turn routing  $R$  that solves  $P$ .

**Notation:** (Worst-case width for  $n \times n$  arrays.) We let  $w(n)$  denote the maximum width of any problem instance defined on an  $n \times n$  array.

**Notation:** (Restriction to  $t$ -turn routings.) We let  $w_t(n)$  denote the maximum of  $w_t(P)$  for any problem instance  $P$  defined on an  $n \times n$  array.

**Notation:** For  $p \neq 1$ , we use the notations  $w(n, p)$  or  $w_t(n, p)$ .

**Remarks:** The reader will be able to distinguish the notations  $w(R)$ ,  $w(P)$ , and  $w(n)$  by the type of the argument.

### Motivation

Our research was motivated by the following intriguing conjecture.

**Conjecture (Thompson):**  $w(n) = w_1(n) = \lfloor \frac{n}{2} \rfloor + 1$ .

This controversial-sounding conjecture states that in the worst-case we need only consider one-turn routings.

On the other hand, it is only requiring that for any problem instance  $P$  there exist a one-turn routing  $R$  for  $P$  such that  $w(R) \leq w(n)$  and not that  $w(R) \leq w(P)$ . (It is not difficult to develop problem instances  $P$  for which  $w(P) = 1$  but  $w_1(P) = \Omega(n)$ .)

### Results

Our major theorems are listed here; proofs and proof sketches are generally given later.

**Theorem 1.**  $\lfloor \frac{n}{2} \rfloor \leq w(n) \leq n$ .

**Proof:** For the lower bound connect  $(i, j)$  to  $(i, n - j)$  for  $1 \leq j \leq \frac{n}{2}$ , and consider the number of wires that must cross from column  $\lfloor \frac{n}{2} \rfloor$  to  $\lfloor \frac{n}{2} \rfloor + 1$ . For the upper bound use any routing in  $\Gamma_1(P)$  for a given instance  $P$ . ■

**Theorem 2.**  $\lfloor \frac{n}{2} \rfloor + 1 \leq w_1(n) \leq \lfloor \frac{n}{2} \rfloor + 2$ .

Furthermore, a one-turn routing  $R$  with  $w(R) \leq \lfloor \frac{n}{2} \rfloor + 2$  can be computed in time  $O(n^3 \log(n))$ .

**Remarks:** Theorem 2 very nearly proves part of Thompson's conjecture. We do not know how to resolve the small difference remaining in Theorem 2. The upper-bound proof involves the development of an elegant algorithm of independent interest for computing a good integral approximation to the solution of a set of linear equalities. The following theorem states the main result used.

**Theorem 3. [The Rounding Theorem]** Let  $A$  be a real-valued  $r \times s$  matrix and let  $\Delta$  be a positive real number such that in every column of  $A$ ,

- (i) the sum of the positive elements is  $\leq \Delta$ , and
- (ii) the sum of the negative elements is  $\geq -\Delta$ .

Let  $\mathbf{x}$  be an  $s$ -vector and  $\mathbf{b}$  and  $r$ -vector such that  $A\mathbf{x} = \mathbf{b}$ . Then there exists an integral  $s$ -vector  $\hat{\mathbf{x}}$  such that

(i) for all  $i$ ,  $1 \leq i \leq s$ , either  $\hat{x}_i = \lfloor x_i \rfloor$  or  $\hat{x}_i = \lceil x_i \rceil$  (i.e.  $\hat{\mathbf{x}}$  is a "rounded" version of  $\mathbf{x}$ ).

(ii)  $A\hat{\mathbf{x}} = \hat{\mathbf{b}}$ , where  $\hat{b}_i - b_i \leq \Delta$  for  $1 \leq i \leq r$ .

Furthermore,  $\hat{\mathbf{x}}$  can be computed from  $A$ ,  $\mathbf{x}$ , and  $\mathbf{b}$  in time  $O(r^3 \log(\frac{s}{\Delta}))$ .

**Remarks:** The Rounding Theorem says that when  $\mathbf{A}$  has only a few small nonzero entries in each column, then we can effectively round  $\mathbf{x}$  to a nearby *integral* point  $\hat{\mathbf{x}}$  while keeping  $\mathbf{A}\hat{\mathbf{x}}$  from increasing very much over  $\mathbf{A}\mathbf{x}$ .

**Theorem 4.** It is NP-complete to determine, given an instance  $P$  of our global routing problem, whether  $w_1(P) \leq \lceil \frac{p}{2} \rceil - 2$ .

**Remarks:** This result is perhaps surprising in view of Theorem 2; the approximation algorithm presented there is remarkably good. The result can also be improved, although we do not include the details in the abstract. In particular, it is also NP-complete to determine whether  $w_1(P) \leq \lceil \frac{p}{2} \rceil - 1$ . When  $p$  is even, it is NP-complete to determine whether  $w_1(P) < \frac{pn}{2}$ . Given the result proved in Theorem 5, this result is as tight as possible.

**Theorem 5.** When  $p$  is even,  $w_1(n, p) = \frac{pn}{2}$ .

**Corollary.**  $\lceil \frac{p}{2} \rceil \leq w_3(n) \leq \lceil \frac{p}{2} \rceil + 1$ .

**Corollary.** When  $p$  is odd,  $p \cdot \lceil \frac{p}{2} \rceil \leq w_3(n, p) \leq \lceil \frac{pn}{2} \rceil + p$ .

**Remarks:** Theorem 5 shows that three-turn routings can yield an improvement (by one). The upper bound proof uses an elegant argument based on finding Eulerian tours in an associated graph.

**Theorem 6.** There is a polynomial time approximation algorithm achieving

$$w(R) \leq O(w(P) \cdot \log\left(\frac{pn}{w(P)}\right))$$

for any problem instance  $P$ .

**Remarks:** The proof of theorem 6 involves a hierarchical bottom-up approach, using a recursion based on  $2 \times 2$  subdivisions. We believe it is possible to reduce the logarithmic term to a constant, but have not yet been able to do so. The result is also valid for  $P$  containing multipoint nets.

**Theorem 7.** If  $n \equiv 2 \pmod{4}$  or  $n \equiv 3 \pmod{4}$ ,  $w(n) \geq \lceil \frac{n}{2} \rceil + 1$ .

**Remarks:** This lower bound extends that of Theorem 2 to handle routings having arbitrarily many turns, in the cases indicated.

**Theorem 8.**  $w_2(n) \leq \lceil \frac{n}{2} \rceil + 1$ .

**Remarks:** This theorem refines the techniques and results of Theorem 5 and its first corollary, moving from three-turn to two-turn nets and improving the upper bound for odd  $n$  by one.

#### Discussion of the Model

Chen *et. al* [CFKNS77] give an excellent overview of how IBM uses algorithms for solving this global routing problem to automatically wire master-slice logic arrays for their System/370 implementations. The model is particularly appropriate for gate-array technologies where each cell might contain a single NAND gate. Fabrication turn-around time can be very small here since wafers can be preprocessed to contain the array of gates and

the only processing required once the logic design is finished is to produce horizontal and vertical wiring on the last two metal layers, to connect the gates together as desired. However, the preprocessing involved usually fixes an upper bound on the value of  $w(R)$  that will be allowed – if all routing channels between gates have width 20 then the routing can not be realized if  $w(R) > 20$ .

As noted earlier, our concern is with “worst-case” values  $w(n)$ ; in practice one would expect “typical” chips to have  $w(P)$  substantially less than  $w(n)$ .

#### Related Work

Burstein and Pelavin [BP83] present an interesting recent “hierarchical” approach to this global routing problem. Much of the earlier algorithmic work (e.g. [HN83]) involved variations on standard shortest-path algorithms, used to route one net at a time. Some probabilistic models have been developed by El Gamal [EG81] to estimate  $w(P)$  under various assumptions about the average distance between the pins on a net, etc., in a typical instance  $P$ . Johnson [J82] gives an overview of the NP-completeness results known in this area.

#### Proof of Theorem 2: One-Turn Routing by Rounding

**Theorem 2.**  $\lceil \frac{n}{2} \rceil + 1 \leq w_1(n) \leq \lceil \frac{n}{2} \rceil + 2$ .

**Proof:** For  $n = 2$ , the lower bound example is easily constructed. (For example, see Figure 3.) For larger values of  $n$ , simply embed the 2-by-2 example in a “width-2 cross” of 0-turn vertical and horizontal nets.

The upper bound is proved using the Rounding Theorem. We first describe how to apply the Rounding Theorem to our routing problem, and then in the next section describe a surprisingly efficient “rounding algorithm”.

We assume for convenience here that  $n$  is even. Let  $x_i$  be a 0–1 valued variable associated with net  $i$  indicating which of the two L-shaped routes will be used. The interpretation is fixed but arbitrary. We assume here that each L-shaped route has *exactly* two wire segments. If both pins for a net lie in the same row or column we assume the two L-shaped routes are distinguished by the inclusion of different zero-length wire segments at their ends. (These are degenerate L-shapes with one leg of the L having zero length.) Each assignment of 0–1 values to  $\mathbf{x} = (x_1, \dots, x_{n^2/2})$  places an easily computed number of wire segments in each row and column. For example, in the problem of Figure 3 the number of wire segments in column 1 is  $(1 - x_1) + x_2$ .

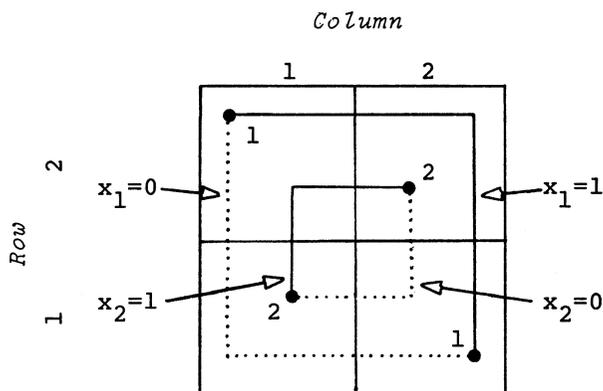


Figure 3

It is then simple to write a set of equations specifying that each row and column will contain exactly  $\frac{n}{2}$  wire segments:

$$Ax = b \quad (*)$$

where  $A$  is a  $(2 \times n) \times (\frac{n^2}{2})$  real-valued matrix. Each variable  $x_i$  will participate in at most four constraints, since its two L-routes affect the wire segment count in at most two rows and two columns. Furthermore, it is easy to check that  $A$  satisfies the conditions of the rounding theorem with  $\Delta = 2$ , since each  $x_i$  will enter two constraints positively and two negatively. Finally, it is easy to see that the vector  $x = (1/2, 1/2, 1/2, \dots, 1/2)$  satisfies the equation  $(*)$ , since each net endpoint will then add  $1/2$  to the wire segment count for its row and column.

Applying the rounding theorem, we infer the existence of a 0-1 valued vector  $\hat{x}$  such that

$$A\hat{x} \leq b + (2, 2, \dots, 2)$$

Except for the claims regarding running times, this proves Theorem 2. ■

### Proof of Theorem 3: The Rounding Algorithm

**Theorem 3.** [The Rounding Theorem] Let  $A$  be a real-valued  $r \times s$  matrix and let  $\Delta$  be a positive real number such that in every column of  $A$ ,

- (i) the sum of the positive elements is  $\leq \Delta$ , and
- (ii) the sum of the negative elements is  $\geq -\Delta$ .

Let  $x$  be an  $s$ -vector and  $b$  and  $r$ -vector such that  $Ax = b$ . Then there exists an integral  $s$ -vector  $\hat{x}$  such that

- (i) for all  $i$ ,  $1 \leq i \leq s$ , either  $\hat{x}_i = \lfloor x_i \rfloor$  or  $\hat{x}_i = \lceil x_i \rceil$  (i.e.  $\hat{x}$  is a "rounded" version of  $x$ ).
- (ii)  $A\hat{x} = \hat{b}$ , where  $\hat{b}_i - b_i \leq \Delta$  for  $1 \leq i \leq r$ .

Furthermore,  $\hat{x}$  can be computed from  $A$ ,  $x$ , and  $b$  in time  $O(r^3 \log(\frac{n}{\epsilon}))$ .

**Proof:** We now describe a "rounding algorithm" that efficiently computes the vector  $\hat{x}$  whose existence is assured by the Rounding Theorem. The input and output parameters are as described in that theorem:

Input:  $A$ ,  $\Delta$ ,  $b$ ,  $x$ .

Output:  $\hat{x}$

The execution time of our rounding algorithm is  $O(r^3 \log(\frac{n}{\epsilon}))$ . In our routing application, we have  $r = 2n$  (one equality for each row or column) and  $s = \frac{n^2}{2}$  (one variable for each net), so the execution time is  $O(n^3 \log(n))$ . This compares favorably with the more usual approach based on shortest paths, which runs in time  $O(n^4)$  to route an  $n \times n$  array.

The steps of our rounding algorithm are:

**Step 1.** [Convert to 0-1 problem.] Replace  $x$  by  $x - x'$ , where  $x'_i = \lfloor x_i \rfloor$  for all  $i$ . Replace  $b$  by  $b - Ax'$ . Solve the modified problem (steps 2 to 3) and then convert back by adding  $x'$  to the  $\hat{x}$  computed and  $Ax'$  to the  $\hat{x}$  computed. Halt.

**Step 2.** [Fast reduction in the number of variables.] This step reduces the number of variables to  $\leq r$  by  $O(\log(\frac{n}{\epsilon}))$  passes through steps 2a-2f.

2a. [Test if done.] If  $s \leq r$ , go to step 3.

2b. [Grouping.] Divide the  $s$  variables into  $(r+1)$  groups, each of roughly the same number of variables. Consider a new problem  $Cy = b$  where  $y$  is an  $(r+1)$ -vector having one element for each group, and  $C$  is  $r \times (r+1)$  matrix.  $C$  and  $y$  are obtained from  $A$  and  $x$  by adding the constraint that the within each group each variable will have the same value. For example, the first column of  $C$  is the sum of the columns of  $A$  corresponding to variables in the first group, and  $y_1$  is the sum of the  $x_i$ 's from the first group.

2c. [Reduce  $C$  to row-echelon form.] Using elementary row operations, convert the  $r \times (r+1)$  matrix  $C$  to row-echelon form, as in Figure 4, (if  $C$  has rank  $r$ ). Note that this operation does not change the null space of  $C$ .

$$C = \left[ \begin{array}{cccc|c} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ 0 & & & & 1 \\ & & & & & \alpha \end{array} \right]$$

Figure 4

2d. [Round.] Let  $s$  be an  $r+1$ -vector in the null space of  $C$ . (This is easy to compute given step 2c.) Let  $\lambda^* = \min\{\lambda \geq 0 \mid y + \lambda \times s \text{ has an integral component}\}$  and let

$$w = y + \lambda^* \times s$$

2e. [Update.] For each variable  $x_i$  in a group  $j$  where  $w_j$  is integral, fix  $\hat{x}_i$  at  $w_j$  and remove  $x_i$  from the problem (set  $b = b - w_j \cdot A_{[r,i]}$ , where  $A_{[r,i]}$  is the  $i$ -th column of  $A$ , delete  $x_i$  from  $x$  and delete the  $i$ -th column of  $A$ .) Set the remaining  $x_i$ 's to their group values  $w_j$ 's.

2f. [Revise group structure.] If now  $s \leq r$ , go to step 3. Else split the largest group into two smaller ones, update  $C$  to reflect the changes in steps 2d and 2e, and return to step 2d.

**Step 3.** [One by one reduction in equalities and variables.] If  $s \leq r$  execute step 3a, else execute step 3b. Repeat step 3 until all variables have been fixed. Then halt; the desired solution has been found.

3a. [*s* ≤ *r*: *Eliminate an equality.*] Find an *i* such that the elimination of equality *i* will not affect the final result (it can be proven that such an *i* always exists in this case). Eliminate this row from matrix **A** and from **b**.

3b. [*Eliminate a variable.*] This is much as in step 2, except we may only eliminate one variable; here each variable is in its own group.

This completes our description of the rounding algorithm. It is not too difficult to verify the claimed running time. ■

**Proof of Theorem 4: NP-Completeness of Optimal One-Turn Routing**

**Theorem 4.** It is NP-complete to determine, given an instance *P* of our global routing problem, whether  $w_1(P) \leq \lceil \frac{n}{2} \rceil - 2$ .

**Proof:** The reduction is from 3-SAT. Given an instance *E* of 3-SAT with variables  $x_1, x_2, \dots, x_q$  and clauses  $c_1, c_2, \dots, c_m$ , set  $n = 14m + 3$  and define the routing problem *P* as follows.

Pins in the rightmost  $7m + 3$  columns of the grid are not included in any net. Any 1-turn routing of *P* can thus have row widths at most  $7m = \lceil \frac{n}{2} \rceil - 2$ . Therefore, we are only concerned with column widths in what follows.

Pins in the leftmost  $7m$  columns but not in the middle 7 rows are paired so that the pin in the *i*th row of the *j*th column is linked to the pin in the  $(n - i + 1)$ st row of the *j*th column. Each of these nets must be routed as a vertical wire, and the question of whether  $w_1(P) \leq \lceil \frac{n}{2} \rceil - 2$  is equivalent to the question of whether the middle 7 rows can be routed with column width 2.

The middle 7 rows of the leftmost *m* columns are used to represent the clauses (one column for each clause). The middle 7 rows of the next  $6m$  columns are used to represent the variables ( $2r_i$  columns for variable  $x_i$  where  $r_i$  is the number of times  $x_i$  appears in *E*). The columns used to represent  $c_j$  and  $x_i$  are shown in Figure 5. The order of the columns from left to right is arbitrary.

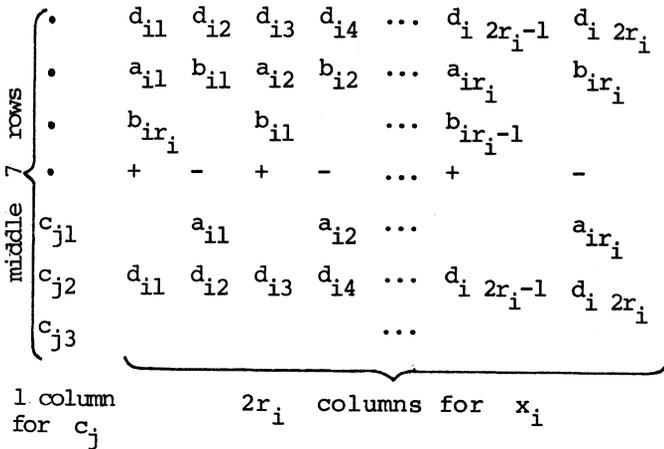


Figure 5

If the *k*th term in clause  $c_j$  ( $k = 1, 2$  or  $3$ ) is  $x_i$ , then any one of the + symbols in the  $2r_i$  columns for  $x_i$  is replaced by  $c_{jk}$ . If the *k*th term in clause  $c_j$  is  $\bar{x}_i$ , then any one of the -

symbols in the  $2r_i$  columns for  $x_i$  is replaced by  $c_{jk}$ . Since  $x_i$  appears  $r_i$  times in *E*, there are always enough +’s and -’s for all the  $c_{jk}$ ’s. The remaining +’s and -’s (as well as the dots) are not assigned to a net. In what follows, we show that the middle 7 rows can be routed with column width 2 if and only if *E* is satisfiable.

Clearly the nets labeled with  $d_{ik}$ ’s must be routed as vertical wires. This leaves only two ways to route the nets labeled with  $a_{ik}$ ’s and  $b_{ik}$ ’s. The two routings correspond in a natural way to the truth value of the associated variable  $x_i$ . The routings are shown in Figure 6.

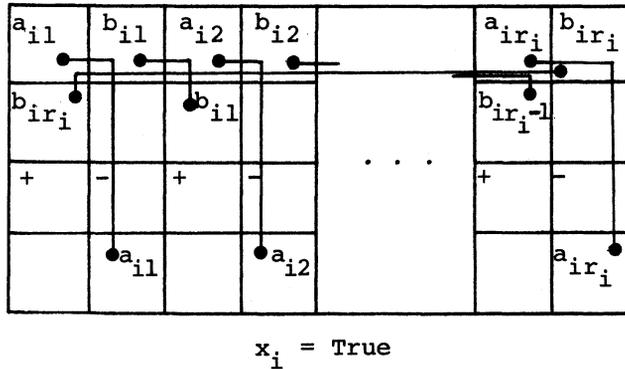
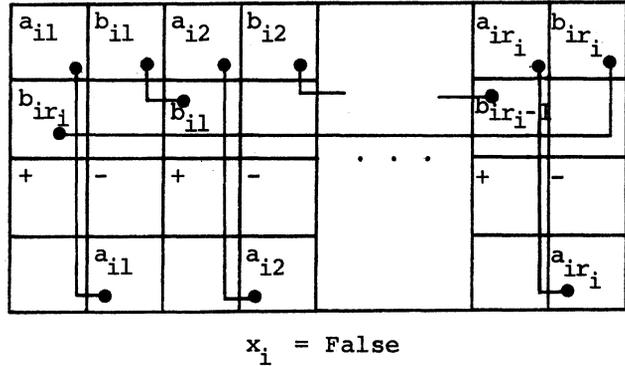


Figure 6

It remains to route the  $c_{jk}$ ’s. It is easily shown that if  $c_{jk}$  corresponds to  $x_i$  where  $x_i$  has a true routing or to  $\bar{x}_i$  where  $x_i$  has a false routing, then net  $c_{jk}$  can be safely routed without using a vertical wire segment in the column for  $c_j$ . This is not the case if  $c_{jk}$  corresponds to  $x_i$  where  $x_i$  has a false routing or to  $\bar{x}_i$  where  $x_i$  has a true routing. In the latter cases, the net for  $c_{jk}$  must include a vertical wire segment in the column for  $c_j$  that passes through the top of the cell containing  $c_{j1}$ . Hence the middle 7 rows can be routed with column width 2 if and only if there is a *k* for each *j* such that  $c_{jk}$  corresponds to  $x_i$  where  $x_i$  has a true routing or to  $\bar{x}_i$  where  $x_i$  has a false routing. This condition is equivalent to *E* being satisfiable. ■

**Proof of Theorem 5: Routing using Eulerian Tours**

**Theorem 5.** When  $p$  is even,  $w_1(n, p) = \frac{pn}{2}$ .

**Proof:** Let each of the cells of the  $n \times n$  array be the vertices of a graph, and connect any two vertices that are connected by a net.

Since  $p$  is even, every vertex will have an even degree.

Thus the edges can be organized into a directed path which is an Eulerian tour, traversing each edge exactly once. (The case that the graph is not connected arises but is easy to handle...)

For each edge  $(i, j) \rightarrow (k, l)$  of the Eulerian tour, we use the an L-shaped route with the horizontal arc first:

$$(i, j) \rightarrow (i, l) \rightarrow (k, l)$$

Since each vertex will have  $p/2$  horizontal arcs leaving it and  $p/2$  vertical arcs entering it, we can route the entire chip with  $pn/2$  tracks in each row or column.

To prove the first corollary ( $p = 1$ ), we group the cells into  $2 \times 2$  squares and apply the above construction for  $p = 4$ .

Then we may need to introduce small (length 1) jogs within each square to get the two horizontal arcs leaving on different rows. This we can do with only one extra track for each row or column, yielding  $\lceil n/2 \rceil + 1$  tracks at most. Here each L-shaped route may have a little tail at each end so a net may have three turns total. ■

**Proof of Theorem 6: Provably Good Routing**

**Theorem 6.** There is a polynomial time approximation algorithm achieving

$$w(R) \leq O(w(P) \cdot \log(\frac{pn}{w(P)}))$$

for any problem instance  $P$ .

**Proof:** (Sketch): Let  $cut(P)$  denote the maximum, over all sub-squares of the  $n \times n$  array, of the number of nets which must cross the border of the square, divided by the perimeter of that square. It is easy to see that  $cut(P)$  is a lower bound on  $w(P)$ .

Divide the chip into squares whose sides have length  $\lambda = cut(P)/p$ . Route these squares independently, in an arbitrary 1-turn manner in width at most  $O(cut(P))$ , routing nets that must leave a square arbitrarily to a point on the perimeter of that square. Then proceed through  $n/\lambda$  levels of bottom-up recursion, at each level pasting together four squares from the previous level in a  $2 \times 2$  pattern, and using at most  $O(cut(P))$  additional width to route all nets that leave the newly constructed square to the perimeter of that square. ■

**Proof of Theorem 7: Improved Lower Bound**

**Theorem 7.** If  $n \equiv 2 \pmod{4}$  or  $n \equiv 3 \pmod{4}$ ,  $w(n) \geq \lfloor \frac{n}{2} \rfloor + 1$ .

**Proof:** Let  $f = \lfloor \frac{n}{2} \rfloor$  and  $c = \lceil \frac{n}{2} \rceil$ . Consider dividing the chip as shown in Figure 7 into four quadrants A, B, C, D, where A is  $f \times f$ , B and C are  $f \times c$ , and D is  $c \times c$ .

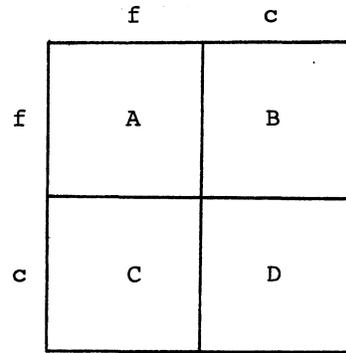


Figure 7

Consider a problem instance where each pin of A is to be connected to a corresponding pin in D, and each pin in B is connected with a pin in C. (If  $c > f$ , the remaining pins in D can be left unattached, or paired off.) Since  $|A| = f^2$  is odd, at least  $\lceil f^2/2 \rceil$  of the wires from A must run through B (without loss of generality – the case for C is symmetric). Thus the perimeter of B will be crossed at least  $(f^2 + 1) + fc$  times:  $(f^2 + 1)$  times for the A-D nets and  $fc$  times for the B-C nets. Since the perimeter of B is crossed by only  $f + c$  channels (rows or columns), at least one of these channels must contain at least

$$\lceil \frac{(f^2 + 1) + fc}{f + c} \rceil = \lceil f + \frac{1}{f + c} \rceil = f + 1 = \lfloor \frac{n}{2} \rfloor + 1$$

wires. ■

**Proof of Theorem 8: Good Two-Turn Routings**

**Theorem 8.**  $w_2(n) \leq \lfloor \frac{n}{2} \rfloor + 1$ .

**Proof:** This is similar to the proof of the first corollary to Theorem 5, except that we group the cells regularly into  $1 \times 2$  rectangles instead of  $2 \times 2$  squares. The Eulerian theorem is applied as before. Finally, the L-shaped routings obtained will have to have at most one tail added to produce the final routing. When  $n$  is even it is easy to arrange the tails without increasing the number of tracks required per channel by more than one. When  $n$  is odd the argument is a little more delicate. Consider labelling each pin either “H” or “V” according to whether the route determined by the Eulerian tour would connect to that pin with a horizontal or vertical segment. Figure 8 shows a labelling that might result for a problem with  $n = 7$ .

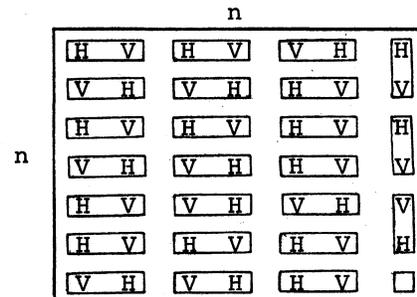


Figure 8

We are guaranteed that each  $1 \times 2$  rectangle contains one "H" and one "V" by the use of the Eulerian tour, and we need to guarantee that each row has at most  $\lfloor \frac{n}{2} \rfloor + 1$  "H"s and that each column has at most  $\lfloor \frac{n}{2} \rfloor + 1$  "V"s. The rows are already OK, if the tiling pattern is like that of Figure 8. To adjust the columns we note that by running a short tail within a rectangle we can effectively move a V "on top of" its neighboring H. We can do this safely only in rows which have a "V" in the rightmost column; otherwise the tail might increase the required channel width. However, there are  $\lfloor \frac{n}{2} \rfloor$  in the rightmost column, so we can always move as many as  $\lfloor \frac{n}{2} \rfloor$  V's out of any column into a neighboring one. Thus we can use the tails to guarantee that no column will have more than  $\lfloor \frac{n}{2} \rfloor + 1$  V's. ■

### Open Problems

We present here some open problems related to the above results. (We hope to be able to answer some of them in our final paper.)

**Open Problem 1:** Is there a constant  $c$  and a polynomial-time global routing algorithm  $A$  such that  $A$  will produce for any problem instance  $P$  a routing  $R$  with  $w(R) \leq c \cdot w(P)$  (i.e. a routing whose width is within a constant factor of optimal)?

**Open Problem 2:** What is  $\frac{w_t(n)}{w(n)}$  for any fixed  $t$ ? Is there a fixed  $t$  for which this ratio equals 1 for all  $n$ ?

**Open Problem 3:** Can the logarithmic factor in the running time of the Rounding Algorithm be eliminated?

**Open Problem 4:** What are other applications of the Rounding Algorithm? (We do know of some ways of applying the algorithm for global routing applications that are more general than the techniques given in this abstract. We suspect that the algorithm may have a large number of useful applications.)

**Open Problem 5:** Let  $cut(P)$  be defined as in the proof of Theorem 6. Is there a constant  $c$  such that  $w(P) \leq c \cdot cut(P)$  for all problem instances  $P$ ? (Note: we can prove that a similar measure computing wire-length within subsquares is linearly related to the  $cut$  measure.)

**Open Problem 6:** Can the additive "+ $p$ " term be improved in the second corollary to theorem 5?

**Open Problem 7:** Develop, empirically or otherwise, a good model of the wiring problem instances that arise in practice.

### References

- [BL83] Bhatt, S. N. and F. T. Leighton, "A Framework for Solving VLSI Graph Layout Problems," *Journal of Computer and System Sciences*, to appear.
- [BP83] Burstein, M., and R. Pelavin, "Hierarchical Wire Routing," IBM T. J. Watson Research Center Report RC9861 (March 1983).
- [CFKNS77] Chen, K. A., M. Feuer, K. H. Khokhani, N. Nan, and S. Schmidt, "The Chip Layout Problem: An Automatic Wiring Procedure," *Proceedings 14th Design Automation Conference* (IEEE, New Orleans, 1977), 298 - 302.

- [EG81] El Gamal, A. A., "Two Dimensional Stochastic Model for Interconnections in Master Slice Integrated Circuits," *IEEE Trans. on Circuits and Systems CAS-28* (February 1981), 127-138.
- [HN83] Hong, S. J., and R. Nair, "Wire Routing Machines - New Tools for VLSI Physical Design," *Proc. IEEE 71* (January 1983), 57 - 65.
- [J82] Johnson, D. S., "The NP-Completeness Column: An Ongoing Guide," *Journal of Algorithms 3* (1982), 381-395.
- [L80] Leiserson, C. E., "Area-Efficient Graph Layouts (for VLSI)," *Proc. 21st FOCS Conference*, (IEEE, October 1980), 270-281.
- [L81] Leighton, F. T., "New Lower Bound Techniques for VLSI," *Proc. 22nd FOCS Conference*, (IEEE, October 1981), 1-12.
- [L82] Leighton, F. T., "A Layout Strategy for VLSI Which is Provably Good," *Proc. 14th ACM STOC Conference*, (ACM, May 1982), 85-98.
- [T79] Thompson, C., "Area-Time Complexity for VLSI," *Proc. 11th ACM Conference*, (ACM, May 1979), 81-88.