Wire Length and Via Reduction for Yield Enhancement *

Venkat K. R. Chiluvuri and Israel Koren†

Advanced Design Technology, Motorola, Austin, TX 78735

†Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003, USA

Abstract

Wire length reduction along with via minimization results in better performance and higher yield for VLSI circuits. In this paper we present a wire length reduction algorithm for channel routing. The results of our algorithm for a set of benchmark examples are presented. The algorithm produces near optimal results for most of the examples. Surprisingly, our algorithm outperforms most of the previously proposed via minimization algorithms as well. Our results show that both wire length and via minimization problems are closely related to each other but their optimal solutions don't necessarily coincide.

Keywords : design for yield, layout synthesis, yield enhancement, channel routing, via reduction, wire length minimization, defect tolerance, VLSI.

1 Introduction

The primary objective of channel routing is to complete the routing in the smallest possible area. Via minimization is the most important secondary objective in several two- and multi-layer channel routers. Several constrained and unconstrained algorithms have also been proposed[8,12,14] for via minimization. In the early routers, the main objective of layer assignment was reducing the number of vias. As the number of layers available for routing has increased, layer re-assignment for performance improvement has assumed significance. For achieving better performance and yield, wire length minimization is also equally important.

Even though wire length minimization has received significant attention in the compaction stage of layout synthesis,[11] it has often been ignored while minimizing the number of vias in the routing stage. If via minimization is pursued very aggressively without paying attention to the attendant wire length increase, intended performance and reliability gains cannot be sustained. For example, if 100 units of additional wire length are introduced just to avoid one via, it can be justified neither from performance nor from yield point of view.[1,4,9] Wire segments are susceptible to open- and short-circuit type faults. A trade-off therefore exists between via and wire length for performance and yield criteria.

The algorithm for yield optimization proposed by Pitaksanonkul *et al.*[13] reduces the overlaps among adjacent wires so as to minimize the defect sensitivity of the layout for short-circuit type faults. In this router only the adjacency information of horizontal tracks is considered. Kuo[10] proposed a new channel routing algorithm for yield enhancement. In this router both layer reassignment and via shifting was used. In this paper, we attempt

*Supported in part by NSF, under contract MIP-9305912.

to minimize both wire length and vias in order to achieve a better yield improvement.

A wire length minimization algorithm for two layer channel routing is proposed in this paper. A channel routing solution with a small area (minimum number of tracks) generated by another router is used as an initial solution and the length of the wire segments in the vertical layer is reduced by rearranging the nets among different tracks. Therefore, our algorithm, in essence, is a post-processor to a channel router for further reducing the wire length. The rest of the paper is organized as follows. In Section 2 we define the problem of wire length minimization and in Section 3 an efficient algorithm is presented. The benchmark results and conclusions are presented in Sections 4 and 5, respectively.

2 Problem Description

In a two-layer channel routing, as the name suggests, two layers are available for routing. One layer is predominantly for horizontal segments and the other layer is for vertical segments. In a routing solution each net is assigned to a track without violating any horizontal and vertical constraints.[15] It is assumed that only one horizontal segment is allowed per net (no dog legs). It should be clarified that our algorithm can process routing with dog legs as well. In our current implementation however, for the convenience of straightforward comparison of wire length reduction we have analyzed only routing solutions without dog legs.

The length of the horizontal segment of a net is determined by its leftmost and rightmost terminals and is independent of its track assignment. However, the length of the vertical segments usually depends on the track assignment. In Figure 1, if net 3 is assigned to track 1 (as in Figure 1b) its vertical wire length is 3, one for each terminal at the top, and if it is assigned to track 2 (as in Figure 1a) its total vertical wire length is 6. For net 6 the total vertical wire length remains 5 irrespective of its track position. Therefore, the vertical wire length depends on the number of terminals in the net and their position, i.e., whether they are from the top or the bottom edge of the channel. We define the weight of a net as the difference between the number of top and bottom terminals. In Figure 1, the weight of net 1 is $+1$ and the weight of net 4 is -2. Unit vertical wire length is defined as the minimum distance (pitch) between two tracks.

Our objective is to reduce the vertical wire length by reassigning nets to various tracks. In the rest of the paper we refer to this vertical wire length reduction as wire length minimization. In Figure 1(a), the total vertical wire length is 38. If the track positions of nets 1 and 3 are interchanged, their vertical wire length is reduced by two units. With similar reassignments of the other nets, as shown in Figure 1(b), the total vertical wire length is reduced to 33 units, resulting in a 13% reduction.

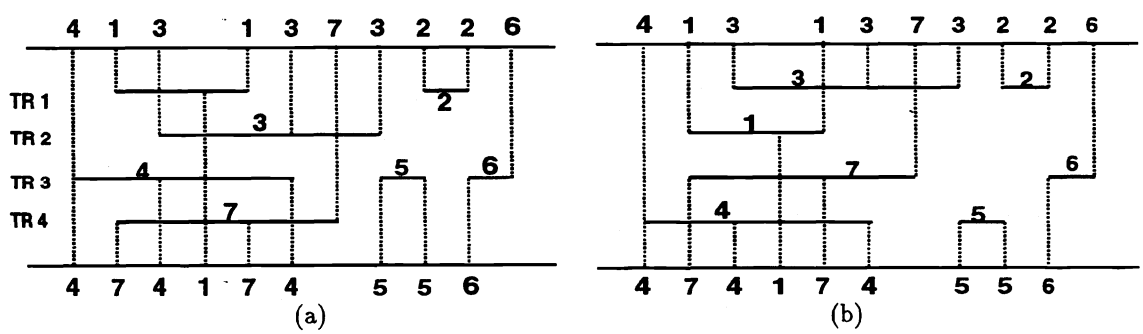


Figure 1: Net-Move and Net-Interchange

3 Wire Length Minimization Algorithm

As shown in Figure 1, if a net with a positive weight (positive net) is assigned to a track close to the top edge its vertical wire length is reduced. Similarly, a net with negative weight (negative net) with a track assignment closer to the bottom edge results in wire length reduction. For a net with zero weight, its vertical wire length is independent of track assignment. The basic idea in our wire length minimization algorithm is to assign as many positive nets as possible to tracks close to the top edge and as many negative nets as possible to the bottom tracks. Though track assignments of zero net weights do not directly contribute to the wire length minimization, they might block some nets in the top and bottom tracks. It is desirable to have those nets assigned to center tracks. Therefore, positive and negative nets are first processed in the decreasing and increasing orders, respectively. When no further improvements are possible, nets with zero weight are processed. In the rest of the paper, for the sake of brevity, we describe all net reassignment procedures with reference to upward direction only.

The algorithm takes an initial channel routing solution generated by another router and reduces the wire length of the vertical layer. In the initialization phase, net weights are calculated and nets are arranged in the decreasing order of their weights. Vertical and horizontal constraints are generated from the net list details. Vertical constraints for each net are represented by its ancestor and descendant nets. Horizontal net constraints are generated from the overlap information of horizontal net segments. Horizontal constraint information is maintained in the form of zone representation and interval graphs.[15] The task of wire length minimization is divided into a set of subtasks for the convenience of implementation and to reduce the runtime complexity. Before introducing the algorithm, the major steps are briefly explained below.

3.1 Net-Move

In channel routing the available track space is normally not filled with horizontal net segments. Track space utilization can be as low as 50% in some routing solutions. For example, in Deutsch's 28-track example the track utility is only 49% and the average track utility of the benchmark examples shown in Table 2 is 66%, i.e., one-third of the track space is empty. We can make use of this empty space for reducing the wire length by shifting nets either to the top or bottom tracks as appropriate. Net shifting is the first step in the algorithm. Nets are processed in the order of decreasing net weight.

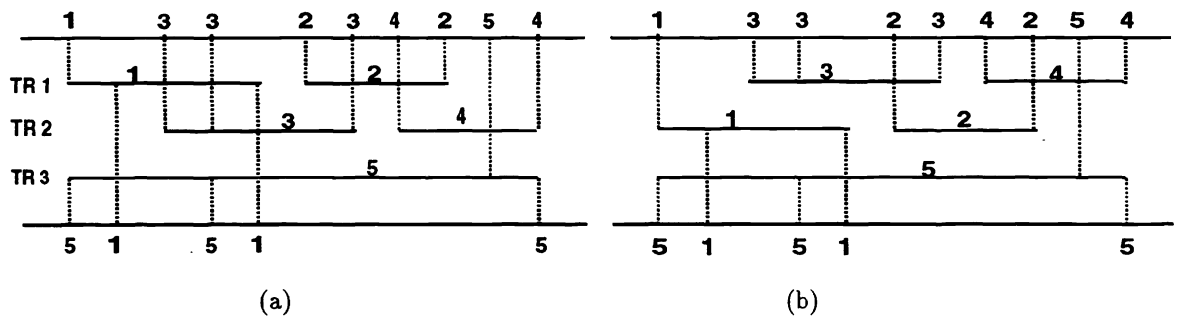
Each positive net requires a search for an empty space in all tracks above its current track, starting from the top edge. If an empty track is found, then in order to move the net to this new track, all ancestors of the net must be in the tracks above this empty track. If this vertical constraint is satisfied then the net is assigned to it. Otherwise the search will continue until the track currently assigned to the net is reached. In Figure 1(b), net 5 is shifted from its current track 4 to track 5 to reduce its vertical wire length by two units.

3.2 Net-Interchange

After moving nets to free tracks, further reduction in wire length is achieved by interchanging the track positions for pairs of nets. If a net is interchanged with a net which has a smaller net weight, wire length can be reduced. For example, by interchanging the track position of net 3 with that of net 1, whose net weight is 1, as shown in Figure 1b, the wire length can be reduced by two units. Therefore, a positive net is a candidate for upward track interchange and a negative net is a candidate for downward track interchange. In the second step of the algorithm, a pair of nets are interchanged in their track positions based on their weight and, the horizontal and vertical constraints.

Once a net is chosen for net-interchange, its left and right end zones are identified. Then, starting from the top edge of the channel, each track is searched for a net with which to be interchanged. The weight of the new net must be less than that of the candidate net, and the left and right ends of the new net must not overlap with other nets in its new track, and vice versa. In addition, the vertical constraints of both the nets should be satisfied. For the net to be moved upward, as mentioned earlier, the track positions of all its ancestors should be above that track. Similarly, for a net to be moved downward, the track positions of all its descendant nets should be below that track. If all the above criteria are satisfied then the nets are interchanged and their track and zone information is updated.

Sometimes two or more shorter nets have to be moved together in order to accommodate one longer net in their current track position. In this case all nets involved must satisfy the net-interchange criteria. There are situations when a single net cannot be interchanged with another net or a set of nets. Two or more nets have to be interchanged simultaneously with two or more nets belonging to another track. In Figure 2a, none of the nets can be interchanged individually with any other net. However, nets 3 and 4 can simultaneously be interchanged with nets 1 and 2. All the above-mentioned net interchanges are performed in the second step of the wire length minimization algorithm.



(a) (b)

Figure 2: Multiple Net-Interchange

3.3 Track-Interchange

In the limiting case all the nets in a track can be interchanged with all the nets belonging to another track, provided the vertical constraints are satisfied. In this case horizontal constraints do not exist among the nets because nets are moved enmasse and the horizontal constraints are already satisfied for the nets belonging to a track. In the third step of the algorithm, this track interchange is attempted in order to speed up the wire length minimization process.

We define the weight of a track as the sum of all the net weights that are currently assigned to it. If the track weights are not in the descending order there may be scope for further wire length reduction. The track weights are examined starting from the top edge of the channel. If the weight of track t is greater than that of track $t-1$, then by interchanging the nets of the corresponding tracks, wire length can be reduced. Even if all the track weights are in decreasing order, it does not necessarily imply that there is no room for further reduction. After track shifting is performed, nets that were not moved earlier due to vertical or horizontal constraints might be able to move to new tracks. Therefore, all the above steps are repeated until none of the allowed net moves are possible for wire length reduction.

3.4 Algorithm

```
From net list information, calculate net-weights;
create vertical and horizontal constraints;
Sort nets; initialize track assignment for nets;
N_p, N_n, N_z = Number of positive weight nets,
negative weight nets and zero weight nets, respectively;
Term-flag = TRUE;
While (Term-flag) do
   Term-flag = FALSE; Move-flag = TRUE;
   Switch-flag = TRUE; Track-flag = TRUE;
   While (Move-flag) do
      for (i = 1; i ≤ N_p; i ++)
         MoveNet-Upwards(i);
   Move-flag = TRUE;
   While (Move-flag) do
      for (i = 1; i ≤ N_n; i++)
         MoveNet-Downwards(i);
   While (Switch-flag) do
      for (i = 1; i ≤ N_p; i++)
        InterchangeNet-Upwards(i);
   Switch-flag = TRUE;
   While (Switch-flag) do
      for (i = 1; i ≤ N_n; i++)
         InterchangeNet-Downwards(i);
    While (Track-flag) do
      for (i = 1; i < Track-count; i++)
         if (track-wt[i+1] > track-wt[i])
            Switch-track[i+1];
   Special-flag = TRUE;
    While (Special-flag) do
      Special-NetMoves;
End;
```

Table 1: Wire length minimization algorithm

In the initialization phase of the algorithm, net weights, track weights, vertical constraint and horizontal compatibility information are generated. Then Net-Move, Net-Interchange and Track-Interchange operations are repeated until no further wire length reduction is possible. Then, a set of special net-move and net-interchange operations are performed. In normal Net-Move operations only nets with positive and negative weights are moved or interchanged in their track positions. In the special Net-Move operations, nets with zero weights are also moved. Similarly, nets are interchanged even when their net weights are equal. These special moves may not directly contribute to wire length reduction, but they may remove some of the vertical constraints among tracks. Once this is done then the regular wire length minimization steps are repeated. This cycle is repeated until there is no room for improvement. Essentially, it is a greedy method in the sense that we do not move or switch any net in such a way that the wire length is increased even temporarily. The algorithm is shown in Table 1. In order to assess the quality of our algorithm, we have formulated the wire length minimization problem as an Integer Linear Program and the results are compared with our heuristic solutions.

Table 2: Results of wire length reduction.

Examples from Yoshimura *et al.*[15]	WL of original solution	WL by ILP	WL by our method	% reduction by ILP	% reduction by our method
Example 1	310	222	222	28.4	28.4
Example 3a	583	511	517	12.3	11.3
Example 3b	818	736	755	10.0	7.7
Example 3c	976	832	841	14.5	13.8
Example 4b	1150	1113	1134	3.2	1.4
Example 5	1309	981	984	25.1	24.8
Deutsch's difficult problem	3486	–	3363	–	3.6
	Average Wire Length Reduction.			13.4	13.0

Table 3: Effect of wire length reduction on defect sensitivity (*example 1*).

Probability of Failure (%)						
Layer	Short-circuit faults			Open-circuit faults		
	Original routing	Optimized routing	% Reduction	Original routing	Optimized routing	% Reduction
Vertical layer	2.24	1.09	51.3	3.53	2.50	29.2
Horizontal layer	2.48	2.10	15.3	3.61	3.55	1.7
Wire Length (*mm*)						
Vertical layer	9.81	6.94	29.3%			
Horizontal layer	11.31	11.09	2.0%			
Vias	57	36	36.8%			

4 Experimental Results

The algorithm has been implemented in C and a set of benchmark examples were analyzed. The results are shown in Table 2. The examples shown in the first column of Table 2 are from Yoshimura and Kuh.[15] In column 2 the total wire length of all net segments in the vertical layer of the original results[15] are shown. Column 3 shows the optimal wire length as obtained from the ILP package and column 4 shows the wire length of our results. Columns 5 and 6 show the percentage reduction in wire length, with reference to the original solutions,[15] obtained by ILP and our method, respectively.

In all the examples wire length reduction is achieved including Deutsch's difficult example. In some examples the wire length reduction is as high as 25% (12% if the horizontal wire length is also taken into account). This reduction in wire length results in better performance due to improvements in *RC* characteristics. This wire length reduction is achieved without increasing the area (same number of tracks). The results are very close to the optimal solutions generated by ILP. The solution for *Deutsch's difficult problem* is obtained in a fraction of a second.

Table 4: Comparison of the via minimization results.

Examples from Yoshimura *et al.*[15]	Original solution	Optimal CVM[12]	The[14]	Our method	Topological routing[8]
Example 1	57	–	–	**36**	–
Example 3a	91	72	66	**59**	42
Example 3b	107	91	78	**78**	69
Example 3c	125	109	103	**92**	83
Example 4b	179	–	–	**116**	86
Example 5	150	114	105	**102**	84
Deutsch's difficult problem	290	234	207	**218**	186

4.1 The Effect of Wire Length Reduction on Yield

To illustrate the effect of wire length reduction on yield due to the proposed algorithm, the benchmark example, *example 1* has been analyzed and the results are shown in Table 3. The layout of the two-layer channel routing of *example 1* has been generated using the MAGIC CAD Tools and *Xlaser*, a yield analysis tool,[7] was used for yield analysis. The wire length of the vertical layer is reduced by 29.3% by the proposed algorithm. The new layout, shown in Figure 3(b) is compared with the original layout solution shown in Figure 3(a). The percentage reduction in the probability of open-circuit faults (29.2%) is almost the same as the percentage reduction in the wire length. However, the percentage reduction in short-circuit faults is much higher (51.3%). This is not totally unexpected because the probability of a short-circuit fault in an element depends on its neighboring elements whereas its open-circuit fault probability is almost independent of its position. The reduction in the probability of short-circuit faults in the horizontal layer is a byproduct of changes in the adjacent tracks. The improvement in the defect sensitivity of the layout due to the reduction in the number of vias has not been evaluated (due to the limitation of the available yield analysis tool).

4.2 Via Reduction

The most interesting byproduct of our algorithm is an incidental via reduction which turns out to be better than or equal to those obtained by most of the via minimization algorithms proposed so far except for the topological channel routing presented by Haruyama *et al.*[8] The comparison of the number of vias used by different algorithms is shown in Table 4. The second column shows the number of vias in the original examples[15] where a reserved layer model is strictly followed. The third and fourth columns show the number of vias used by the Optimum CVM algorithm[12] and another via minimization algorithm by The *et al.*,[14] respectively. The fifth column shows our results and the last column shows the best known via reduction results.[8] The number of vias by our method for *example 3a* and *example 3c* are 11% less than those obtained by The *et al.*[14] However, the number of vias is larger by 20%, on an average, when compared to the best known results.

After analyzing the wire length and via reduction results of our algorithm, we can conclude that up to a point, reduction of the wire length also contributes to the reduction of number of vias. Take for an example, a net with the largest number of terminals from the top edge of the channel. For minimizing the wire length, the top-most track is assigned to it, if all the vertical constraints are satisfied. For a net in the top-most track, vias of the terminals of the same side are automatically eliminated. For reducing the wire length, as many nets as possible are assigned to top and bottom-most tracks. In this process we are also increasing the number of vias that can be eliminated. We suspect this is the major contributing factor to the reduction in the number of vias. Another reason for via reduction is that the top and bottom tracks are filled as much as possible. Therefore, for the nets that are assigned to center tracks, vias can be eliminated, because the chance of net crossing is greatly reduced.

After achieving near-minimum wire length, it is very simple to attempt further reduction in the number of vias. For example, in *example 3b*, vias can be reduced from 78 to 72 by increasing the wire length by six units only. In several examples the number of vias can be reduced close to that reported by Haruyama *et al.* by increasing the wire length by about 5%.

In most of the technologies yield losses due to via/contact failures are significant.[4] Therefore, via reduction achieved by the proposed algorithm results in significant improvement in the defect sensitivity of the circuits. The improvement in yield due to reduction in wire length during compaction stage was reported in Chiluvuri *et al.*[2] on the above benchmark examples. The defect sensitivity is reduced by 8%. We expect a 10-15% reduction in the defect sensitivity due to the wire length and via reduction achieved by the proposed algorithm.

5 Conclusion

An efficient wire length reduction algorithm for yield enhancement is presented for two-layer channel routing. In the process of wire length reduction our algorithm also has outperformed most of the previously proposed via minimization algorithms. Our results show that both wire length minimization and via reduction problems are closely related to each other. In the solutions obtained by our wire length reduction algorithm, which are very close to the optimal solutions, there is scope for further reduction in the number of vias without increasing the wire length. It is safe to conclude that we should not solve these problems in isolation from one another. Via and wire length minimization are two facets of the same problem from the yield, performance and reliability points of view.

6 REFERENCES

[1] V. K. R. Chiluvuri and I. Koren, "Layout Synthesis Techniques for Yield Enhancement," *IEEE Trans. Semiconductor Manufacturing*, pp. 178-187, May 1995.

[2] V. K. R. Chiluvuri and I. Koren and J. L. Burns, "The Effect of Wire Length Minimization on Yield," *IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Systems*, pp. 97–105, 1994.

[3] V. K. R. Chiluvuri, "Layout Synthesis Techniques for Yield Enhancement," *PhD thesis, ECE Department, University of Massachusetts*, Amherst, 1995.

[4] R. S. Collica *et al.*, "A Yield Enhancement Methodology for Custom VLSI Manufacturing," *Digital Technical Journal,* Vol. 4, No. 2, pp 83-99, Spring 1992.

[5] R. B. Fair, "Challenges to Manufacturing Submicron, Ultra-Large Scale Integrated Circuits," *Proceedings of IEEE,* Vol. 78, No. 11, pp. 1687-1705, November 1990.

[6] S. C. Fang *et al.*, " Via Minimization with Associated Constraints in Three-Layer Routing Problem," *Proc. of the ISCAS,* pp. 1632-1635, 1990.

[7] J. P. Gyvez and C. Di, "IC Defect Sensitivity for Footprint-Type Spot Defects," *IEEE Trans. Computer-Aided Design,* Vol. 11, pp. 1177-1197, May 1992.

[8] S. Haruyama, D. F. Wong and D. S. Fussell, "Topological Channel Routing," *IEEE Trans. Computer-Aided Design*, Vol. 11, No. 10, pp. 1177-1197, October 1992.

[9] D. A. Joy and M. J. Ciesielski, "Layer Assignment for Printed Circuit Boards and Integrated Circuits," *Proceedings of IEEE*, Vol. 80, No. 2, pp. 311-331, February 1992.

[10] S. Y. Kuo, "YOR: A Yield-Optimizing Routing Algorithm by Minimizing Critical Areas and Vias," *IEEE Trans. Computer-Aided Design*, Vol. 12, No. 9, pp. 1303-1311, September 1993.

[11] G. Lakhani and R. Varadarajan, "A Wire-Length Minimization Algorithm for Circuit Layout Compaction," *Proc. of the ISCAS,* pp. 276-279, 1987.

[12] N. J. Naclerio, S. Masuda and K. Nakajima, " Via Minimization for Gridless Layouts," *Proc. of the 24th ACM/IEEE Design Automation Conf.,* pp. 159-165, 1987.

[13] A. Pitaksanonkul *et al.,* "DTR: A Defect-Tolerant Routing Algorithm," *26st IEEE Design Automation Conference,* pp. 795-798, 1989.

[14] K. The, D. F. Wong and J. Cong, "Via Minimization by Layout Modification," *Proc. of the 26th ACM/IEEE Design Automation Conf.,* pp. 799-802, 1989.

[15] T. Yoshimura and E. S. Kuh, "Efficient Algorithms for Channel Routing," *IEEE Trans. Computer-Aided Design,* Vol. 1, No. 1, pp. 25-35, January 1982.

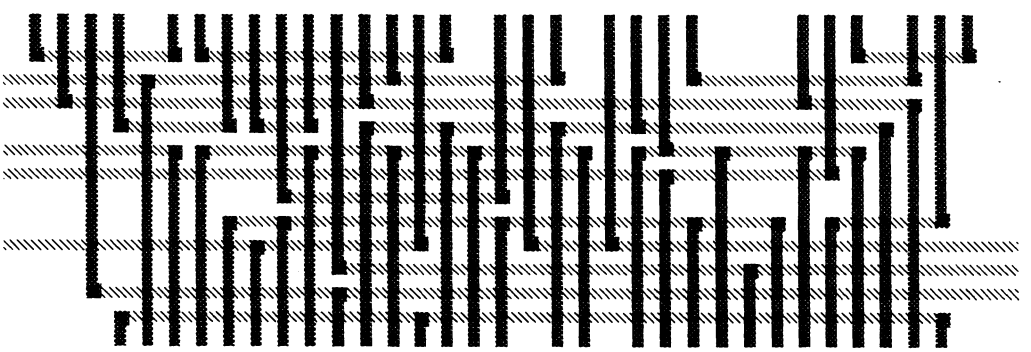


Figure 3(a): Layout of original routing of *example 1* (wire length of metal-2 layer: 9.81mm, 57 vias)

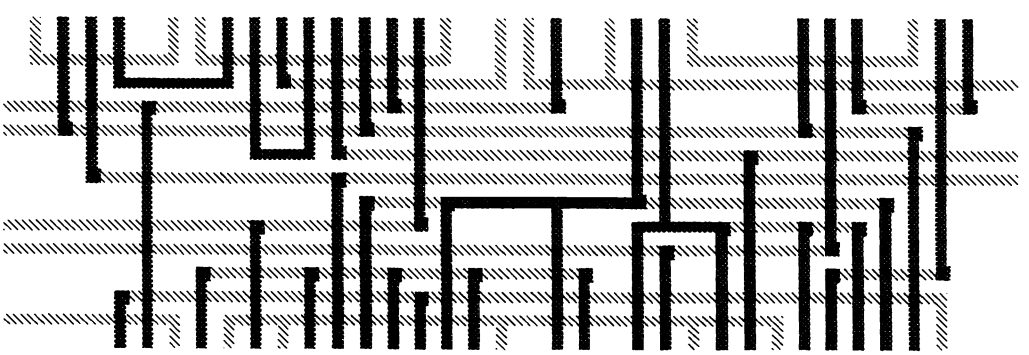


Figure 3(b): Layout of optimized routing of *example 1* (wire length of metal-2 layer: 6.94mm, 36 vias)