Exact Algorithms for Coupling Capacitance Minimization by Adding One Metal Layer *

Hua Xiang
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
huaxiang@us.ibm.edu

Kai-Yuan Chao
Intel Corporation
Hillsboro, Oregon 97124
kchao@ichips.intel.com

Martin D. F. Wong
ECE Dept, UIUC
Urbana, IL, 61801
mdfwong@uiuc.edu

Abstract

Due to the rapid development of manufacturing process technology and tight marketing schedule, the chip design and manufacturing always work toward an integrated solution to achieve enhanced product for fast time-to-market and higher niche profit. For high-end "high-volume" products, one good option of further improving chip performance is to add extra metal layers based on an existing design after all easy circuit fixes and process tricks are already applied. This strategy has recently been applied by main integrated device manufacturers. Contrast to most low volume ASIC, additional metal layer cost is low due to cost averaging over huge volume (e.g., millions per week for x86 mainstream microprocessors). In this paper, we address NLM (New Layer Migration) problem which eliminates coupling capacitance violations for speed push in a given routing solution by migrating some wire segments to a newly inserted metal layer under commonly used metal filling post process for manufacturability and coupling control. We first propose an exact linear-time algorithm to judge whether a feasible solution exists or not. Then we present a provably optimal algorithm to eliminate as many coupling violations as possible. At the same time, the total coupling capacitance on both metal layers is minimized. The time complexity of the algorithm is $O(n^{\frac{3}{2}} \log n)$. Finally an LP approach is presented as post processing to adjust segment positions when the two layers have layer-dependent design spacing rules.

Category: B.7.2 [Integrated Circuits]: Design Aids - Placement and routing; J.6 [Computer Applications]: Computer-Aided Engineering - Computer-aided design

Terms: Algorithms, Design, Performance

Keywords: Max-cut, Capacitance coupling, Layer migration

1. Introduction

Semiconductor industry can be divided into several different areas based on product applications and their market size or profitability. The major revenue and profit of semiconductors are generated by high-volume products such as DRAM and microprocessors. For these products, the custom design flows and integrated manufacturability are commonly seen in industry. The initial design stage of these products usually is longer than ASIC where most published CAD optimization algorithms focus on. However, most manufactured products for these high-volume products came from proliferation of initial design in the modified manufacture processes. For example, Intel Pentium II/III are proliferated from P6 initial design and all AMD XP's are based on K7 initial design.

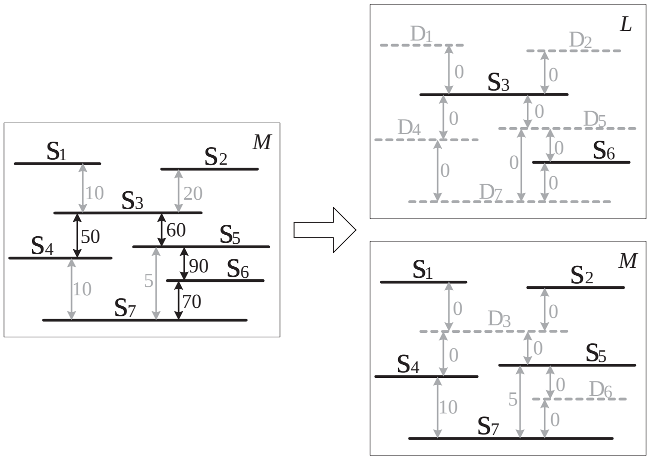
Proliferation, a process of making a new design by modifying an existing product, such as product upgrade for speed push, saves design efforts from designing a totally new chip and produces main revenue from the initial design. It is a highly constrained design optimization based on an existing design with tight design schedule due to time-to-market consideration. Although some relative easy process tricks like fixing speed path, increasing voltage, tuning process recipes, and so on can be applied, the chip performance may still not meet the speed requirement.

One possible option to tune the chip performance higher is process shrink (e.g., from $0.18um$ to $0.13um$). But usually the existing design needs to be re-designed for compaction and it takes long design time since all metal layers have different parasitic characteristics from the original manufacturing process.

An alternative option is to add extra manufacturing cost, such as adding metal layers, to shorten the time-to-market used in major high-end microprocessors [13, 4]. For adding single layer, the newly added metal layer is right above the target routing layer. Usually it has a similar process and parasitic characteristic to the existing target routing layer which helps to save the whole manufacturing process generation change. Also before re-design effort is completed or justified, it is always an option to shift existing design to a new process that has one extra metal layer than the existing manufacture process to perform a quick simulation of possible performance gain (or decide which layer or layers to be added). For high-volume products, the manufacturing cost is quite low (e.g., the die cost is about $10 \sim 20$ cents/ mm^2.) [17]. Especially, different mask layers have different resolution requirement, and upper metal layers are comparatively crude [18]. Therefore masks for upper metal layers are much cheaper than many base layer masks [16, 12]. Although an extra metal layer causes additional manufacturing cost, the introduced cost is tiny after it is amortized over large volume [2, 17]. And the addition of metal layers on existing design is already applied by main integrated device manufacturers [13, 4]. And for high-end niche design, the cost can be justified by the high profit margin of products. Meanwhile, the shortened design time helps keep the advantage of the products in market competition.

On the other hand, since the decision window is very limited, the utilization of the newly added metal layer is a challenging task for design re-optimization. In this paper, we address the NLM (New Layer Migration) problem for the re-assignment optimization of the target routing layer in order to minimize coupling capacitance af-

*This work was partially supported by the National Science Foundation under grant CCR-0306244.



(a) (b)

Figure 1: (a) A routing design has coupling violations. The upper coupling bounds for all segment pairs are 40, while the coupling between s_3 and s_4, s_3 and s_5, s_5 and s_6, s_6 and s_7 is larger than the given bound. (b) An optimal solution. Two segments s_3 and s_6 migrate to the new layer. The gray dotted lines are metal-filled wires to maintain the metal density.

ter adding a new routing layer while keeping the same interconnect topology with all other metal layers untouched. Since the new metal layer is added for alleviating coupling capacitance, it has the same routing direction as the original metal layer where signal wires are routed tightly and cross coupling capacitance contribution in parasitic is high. This is different from conventional process node adventure (e.g., from 0.18*um* to 0.13*um*) where an extra top metal layer is usually added with the routing direction orthogonal to its lower metal layer. In particular, we consider the metal-fill process that is commonly used as a post-design process to increase manufacturability [14, 15] and interconnect coupling [9] if hooked on power mesh. We assume the newly added layer has a similar metal-fill requirement as the original metal layer. Metal fill in high-end design are very dense due to smaller pitch in deeper submicron process.

For convenience, we assume a new metal layer L is added above layer M which is used for horizontal tracks. Figure 1 illustrates an example. On M, there are 7 wire segments. Suppose the required coupling bounds for all segment pairs are 40. The numbers in Figure 1 are the coupling capacitance between two segments. As we notice that the coupling between s_3 and s_4, s_3 and s_5, s_5 and s_6, s_6 and s_7 all exceeds the required bound. After adding a new metal layer L, we want to move some segments to L so that no coupling violations exist in the new design. At the same time, the total coupling of the two layers is minimized. Since wire segments are moved to the new layer, some metal-filled wires are inserted on both layers in order to maintain metal density. Figure 1 (b) illustrates an optimal solution. Two segments s_3 and s_6 migrate to the new layer and the new design has no coupling violations. The gray lines represent metal-filled wires and they cause no coupling effect to other segments. On the other hand, if we move any other set of segments to the new layer, either coupling violations still exist or the new design has larger total coupling than that of the solution in (b).

For short, we say a solution is feasible if there are no segment pairs such that the coupling between the two segments exceeds the required bound. If the coupling capacitance between two segments is larger than the required bound, then one and only one segment must migrate to the new layer. It is easy to solve one pair of segments which has coupling violation. However when several pairs are involved, unwise choice of migrating segments may lead to infeasible solutions. In some cases, there is even no feasible solution to a given NLM problem. Figure 2 shows an example. There are 5 segments on M and the numbers in the figure are the coupling capacitance between two segments. Suppose the required coupling upper bounds for all segment pairs are 40. Therefore, segment pairs (s_1,s_2), (s_1,s_3), (s_2,s_4), (s_4,s_5) and (s_3,s_5) violate the coupling requirement. Then one and only one segment in each pair should migrate to the new layer. However, no feasible solution can be found to this NLM problem. For instance, s_1 is moved to the new layer, then s_2 and s_3 have to stay on the original layer. Since s_2 is on M, s_4 should be placed on the new layer and this makes s_5 on M. On the other hand, s_3 is on M and s_5 have to migrate to the new layer in order to resolve the coupling violation between s_3 and s_5. This leads to contradiction.

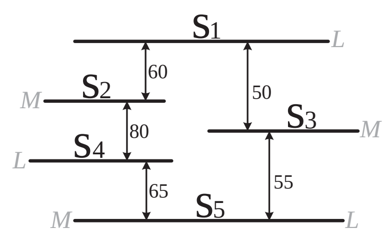


Figure 2: An NLM problem which has no feasible solution.

In this paper, we first present an exact linear time algorithm to judge if a feasible solution exists or not. Then we present another optimal algorithm which not only eliminates as many coupling violations as possible, but also minimizes the total coupling capacitance of both layers. The running time is $O(n^{\frac{3}{2}}\log n)$, where n is the number of segments on M. Fast and optimal algorithms also propose a useful scheme to search which layer to chose by running the program on each layer and making cost-result trade-off management decision. Furthermore, we consider the case when the new layer and the original layer have different design spacing rules. We propose an efficient LP approach to adjust the positions of segments on L to satisfy design spacing requirement while minimizing changes at the same time.

2. New Layer Migration (NLM)

In general, each segment has coupling effects to all other segments. However, the coupling capacitance decreases dynamically if the segment is out of the neighborhood of the other segment [11, 19]. Therefore, we only consider the coupling capacitance between two neighboring parallel wires, and the coupling capacitance can be calculated by the following formula:

$$c = \alpha \frac{length}{distance^{\beta}}$$

where β is a constant approximate to 2. Furthermore, after some wire segments are moved to the new layer, metal-filled wires are inserted at the corresponding positions in order to maintain the metal density. These metal-filled wires have no coupling effects to other wire segments. Therefore, the adjacency of two wire segments is not changed if two segments are still on the same layer, i.e., the coupling capacitance between two wires on the same layer is not changed, and no new capacitive coupling will be introduced after metal-filled wires are inserted.

Since NLM is actually a non-trivial re-assignment optimization problem due to added metal layer based on the existing completed design, detailed timing verification and noise verification are available for all signals. Thus, from the evenly distribution or from some budgeting heuristic (e.g., a percentage of reduction of existing coupling capacitance of wires belonging to critical nets), for each pair of segments, there is a coupling capacitance upper bound that designers want to achieve based on new performance target.

Suppose there are n segments $S = \{s_1, ..., s_n\}$ on M. Let c_{ij} denote the coupling capacitance for the pair of segments (s_i, s_j) $(i, j \in [1..n])$, and b_{ij} be its coupling capacitance upper bound. For a segment pair (s_k, s_l), if $b_{kl} < c_{kl}$, we say the two segments violate coupling requirement. Define coupling violation set $R = \{(s_k, s_l) | b_{kl} < c_{kl}, k \in [1..n], l \in [1..n]\}$.

In order to solve the coupling problem, one new metal layer L is inserted above M. Our target is to decide which segments on M should be moved to L so that the number of coupling violations in the new design is minimized. At the same time, the total coupling capacitance of both layers (M and L) is also minimized.

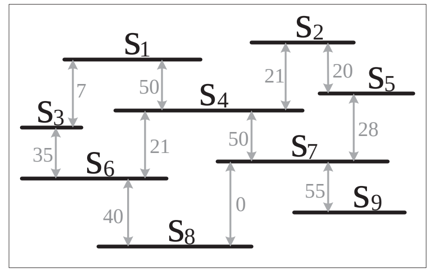
3. NLM Algorithms

In this section, we first introduce how to construct a "coupling graph" which represents the coupling between segments. Based on the coupling graph, we propose two optimal NLM algorithms. The Fast-NLM algorithm determines which segments should migrate to the new layer in linear-time so that the new routing design has no coupling violations as long as a feasible solution exists. Then another optimal Max-Cut-NLM algorithm is presented to eliminate as many coupling violations as possible while minimizing the total coupling capacitance of both layers at the same time.

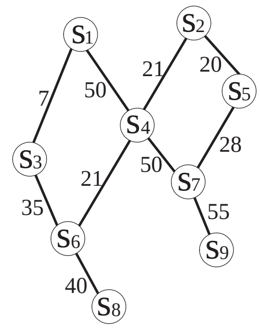
3.1 Coupling Graph

For convenience, a horizontal segment can be represented as (x_1, x_2, y) where (x_1, y) and (x_2, y) are the coordinates of the two end points of the segment. For any two segments $A(x_{a1}, x_{a2}, y_a)$ and $B(x_{b1}, x_{b2}, y_b)$, let $[\bar{x}_1, \bar{x}_2] = [x_{a1}, x_{a2}] \cap [x_{b1}, x_{b2}]$ and $(y_a < y_b)$. If $[\bar{x}_1, \bar{x}_2] \neq \phi$ and no other segments overlap with the rectangular area $(\bar{x}_1, y_a, \bar{x}_2, y_b)$ where $(\bar{x}_1, y_a)$ and $(\bar{x}_2, y_b)$ are the coordinates of its bottom-left corner and up-right corner respectively, then the rectangular area is called "C-Region".

According to our coupling capacitance model, only adjacent segments may have coupling effects. To represent the coupling capacitance between segments, we construct an undirected graph "coupling graph" as follows.



(a)



(b)

Figure 3: (a) A routing solution; (b) The corresponding "coupling graph".

Each horizontal wire segment is represented by a node. Without misunderstanding, we use the same notation for segments and nodes, and refer to a node as its corresponding segment, vice versa. For any two segments u and v, if the coupling effect is non-zero, one edge (u, v) is added to the graph. Figure 3 shows an example.

The number of nodes in a coupling graph G is n (the total number of segments on M). For any two nodes in the graph, if there is an edge e between them, the two segments must be adjacent and a "C-Region" must exist. Since no other segments go across the "C-Region", no edges cross edge e. Therefore, G is a planar graph. According to the property of planar graph, the number of edges in G is $O(n)$.

Lemma 1. *The corresponding coupling graph of a given NLM problem is a planar graph.*

3.2 Fast-NLM Algorithm

In this section, we propose an exact linear-time Fast-NLM algorithm to test if we can find a feasible solution to the given NLM problem.

According to the coupling capacitance model, if one segment migrates to the new layer, the remaining segment pairs are not affected in terms of coupling capacitance since the distance between two segments on M are not changed. At the same time, if two segments are moved to the new layer, the coupling capacitance between them is still the same as that on M. Therefore, no new coupling capacitance between any two segments will be introduced when some segments are moved to L.

To check if a feasible solution exists to a given NLM problem, we only need to consider the segments involved in coupling violations. Similar to the construction of the coupling graph, we can build a "violation graph" $\bar{G} = (\bar{V}, \bar{E})$. If a segment appears in a segment pair (s_k, s_l) ($(s_k, s_l) \in R$), it is represented by a node. And the segment pair (s_k, s_l) is represented by an edge connecting two nodes s_k and s_l. Since $\bar{G}$ is just a subgraph of the coupling graph, the number of nodes and edges is also $O(n)$.

For a violation graph, if we can find two independent sets $\bar{S_1}$ and $\bar{S_2}$ such that $\bar{S_1} \cup \bar{S_2} = \bar{V}$ and $\bar{S_1} \cap \bar{S_2} = \phi$, then we can conclude that a feasible solution exists to the given NLM problem. This is equivalent to judge if a violation graph is bipartite or not. As illustrated by the pseudo code of FAST-NLM algorithm, if a node detects any node in its neighborhood already labeled with the same layer, the graph is not bipartite, otherwise it is. At the same time, if $\bar{G}$ is a bipartite graph, Fast-NLM returns a partition recorded in an array *layer*.

```
Algorithm Fast-NLM(S, R)
Construct violation graph Ḡ(V̄, Ē)
Initialize queue Q = {s1}
layer[s1] = 1
For each node si in V̄ − {s1}
      layer[si] = −1

While Q is not empty
      u = DEQUEUE(Q)
      For each neighbor v of u
         If (layer[v] == −1)
         Then layer[v] = (layer[u] + 1)  mod 2
               ENQUEUE(Q, v)
         Else If (layer[v] ≠ (layer[u] + 1)  mod 2)
            Then return "No Solution"
EndWhile
Return layer
```

The construction and initialization of the violation graph take $O(n)$ time. For each edge in $\bar{G}$, it is visited at most once. Therefore the judgment can be made in linear time $O(n)$.

Theorem 1. *Fast-NLM algorithm can exactly judge whether a feasible solution exists or not to a given NLM problem in $O(n)$ time where n is the number of wire segments on the original layer.*

3.3 Max-Cut-NLM Algorithm

NLM problem not only requires to remove coupling violations, but also tries to reduce the total coupling of both layers. Therefore, if the coupling between two segments is large, it is preferable to separate the two segments to different layers.

Each NLM problem corresponds to a coupling graph G. To handle coupling violations and to minimize total coupling in a consistent way, we use a large "weight" w to adjust the edge cost in G. For any two segments (s_k, s_l) ($(s_k, s_l) \in R$) whose coupling capacitance is c_{kl}, let the edge cost be $w \cdot c_{kl}$. w is set to be a very large number such that $w \cdot c_{kl}$ overwhelms the coupling capacitance of all segment pairs. For example, w can be defined as $1 + \lceil \sum c_{ij} / \min\{c_{kl}\} \rceil$ where $\sum c_{ij}$ is the total coupling capacitance among segments on M and $\min\{c_{kl}\}$ is the minimum coupling capacitance of a segment pair in R. In other words, the cost of an edge (s_k, s_l) ($(s_k, s_l) \in R$) is even larger than the total coupling capacitance in the original design. As to segment pairs not belonging to R, just use the coupling capacitance as the edge cost. Then an NLM problem can be transformed to a Max-Cut problem. Max-cut problem looks for a bipartition of a graph so that the total cost of edges connecting the two node sets is as large as possible. By applying Max-Cut algorithm, we can partition the segments into two sets such that the total cost of edges inside two sets is minimized. Figure 4 illustrates an example.

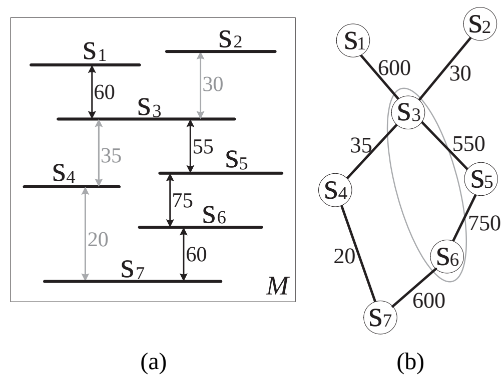


Figure 4: (a) An NLM problem; (b) Node partition obtained by Max-Cut.

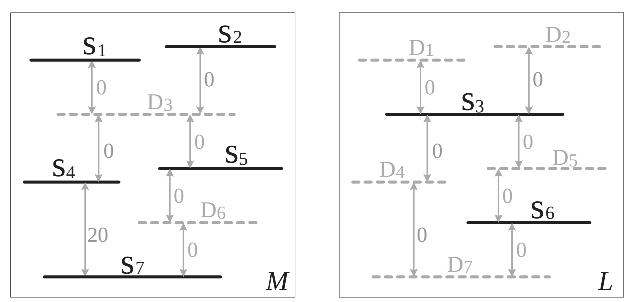


Figure 5: An NLM solution by moving s_3 and s_6 to L

In Figure 4 (a), there are 7 segments and 4 coupling violations. (b) is the corresponding coupling graph supposing $w = 10$. For edges related to coupling violations, the edge cost is 10 times of the coupling capacitance. The gray ellipse shows the partition result after applying Max-Cut on the coupling graph. Figure 5 is an NLM solution derived from the Max-Cut partition in Figure 4. This solution does not have coupling violations.

The Max-Cut-NLM algorithm can be summarized as follows. S is the segment set and R is the coupling violation set. w is the weight and *threshold* is a constant which is used to judge if an output is a feasible solution or not. The value of *threshold* could be $w \cdot \min\{c_{kl}\}$ where c_{kl} is the coupling capacitance of (s_k, s_l) ($(s_k, s_l) \in R$).

```
Algorithm Max-Cut-NLM(S, R, w, threshold)
Construct coupling graph G
For each edge e(u,v) in G
    If (u,v) ∈ R
    Then cost[e] = w · coupling(u,v)
    Else cost[e] = coupling(u,v)
Apply Max-Cut algorithm on G
Move segments in one subset to L
If (total_coupling ≥ threshold)
Then return "No Feasible Solution"
```

If there is a feasible solution to the given NLM problem, any two segments u and v which are involved in coupling violations will be separated into different subsets since the cost of edge (u,v) is huge. On the other hand, if there is no feasible solution, at least one subset includes an edge whose cost is large. By comparing the total cost of the two subsets with *threshold*, it is easy to judge if the new design still has coupling violations or not. Meanwhile, if no feasible solution exists, Max-Cut-NLM algorithm always returns a solution which resolves coupling violations as many as possible and minimizes the total coupling at the same time.

The graph construction and edge cost assignment take $O(n)$ running time. The main part of Max-Cut-NLM algorithm is Max-Cut algorithm. A general Max-Cut problem is an NP-complete problem [6, 10]. However, when the graph is a planar graph, the Max-Cut problem can be translated into a maximum weighted matching problem [1, 3, 8], and it can be solved in $O(n^{\frac{3}{2}}\log n)$ time [8]. Therefore, the Max-Cut-NLM algorithm can be accomplished in $O(n^{\frac{3}{2}}\log n)$ time.

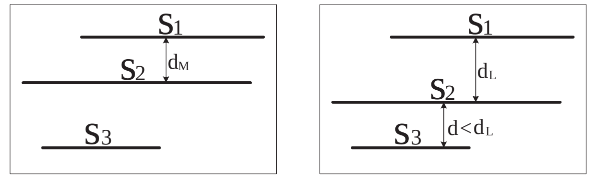
Theorem 2. *The Max-Cut-NLM algorithm optimally solves NLM problems in $O(n^{\frac{3}{2}}\log n)$ time where n is the number of wire segments on the original layer.*

4. Adjustment for Layer-Dependent Spacing Rules

Although the newly added layer has similar process and parasitic characteristic as the original layer, in some cases, the difference in design specification may still exist. In this section, we consider how to adjust segment positions when the two layers have different design spacing rule requirements.

Suppose the design spacing rules for M and L are d_M and d_L respectively ($d_M < d_L$). Then when some segments migrate to L, they may not satisfy the design spacing rule. For example, if the distance of two segments A and B is d_M and both of them migrate to the new layer, the spacing between A and B is still d_M while the design spacing rule for L is d_L. If we simply move one or both segments up/down to increase the spacing between them, other segments may also be affected. Figure 6 shows an example. Segments s_1, s_2 and s_3 are three segments migrating to the new layer. And the spacing between s_1 and s_2 is d_M which is smaller than the design spacing rule d_L. Then s_2 is moved down so that the spacing requirement between s_1 and s_2 is satisfied. But this causes the spacing between s_2 and s_3 less than d_L too.

Once some segments migrate to the new layer, it is necessary to adjust the spacing in order to eliminate spacing violations. At the same time, we hope the adjustment is as little as possible.



(a) (b)

Figure 6: (a) Three segments migrate to the new layer. But the spacing between s_1 and s_2 is less than d_L. (b) s_2 is moved down in order to increase the spacing between s_1 and s_2, but this makes the spacing between s_2 and s_3 less than d_L.

Suppose there are m segments $\{a_1,\ldots,a_m\}$ on L. Let t_i be the original y-coordinate of a_i, and y_i be its new position. In order to restrict that the new position of a segment is not too far away from its original position, we require $|y_i - t_i| \le T$ where T is a given constant, i.e., each segment can move only within a range $[t_i - T, t_i + T]$. According to the above requirements, we have the following formulation:

$$\min \sum_{i=1}^{m} |y_i - t_i|$$
$$\begin{aligned} y_i - y_j &\ge d_L, \quad \forall i,j, \quad t_i > t_j \\ |y_i - t_i| &\le T \\ y_i &\ge 0, \quad i = 1,\ldots,m \end{aligned}$$

Let $z_i = |y_i - t_i|$. And $|y_i - t_i| \le T$ can be written as $-T \le y_i - t_i \le T$. Then the above formulation can be transformed to a standard linear programming problem as follows.

$$\min \sum_{i=1}^{m} z_i$$
$$\begin{aligned} y_i - y_j &\ge d_L, \quad \forall i,j, \quad t_i > t_j \\ y_i - t_i &\le T \\ y_i - t_i &\ge -T \\ y_i - t_i &\le z_i \\ y_i - t_i &\ge -z_i \\ y_i,\ z_i &\ge 0, \quad i = 1,\ldots,m \end{aligned}$$

It is well known that the above LP problem can be solved in polynomial time [5, 7].

5. Experimental Result and Conclusion

In this paper, we address the NLM problem which solves coupling violations on one layer by adding an extra metal layer. We first propose an exact linear-time algorithm to judge if a feasible solution, which has no coupling violations, exists or not to a given NLM problem. Then we present another optimal algorithm to determine which segments should migrate to the new layer so that the number of coupling violations is minimized. At the same time, the total coupling capacitance in the new design is minimized. The time complexity of the algorithm is $O(n^{\frac{3}{2}}\log n)$. Figure 7 (a) shows a test file with 281 wire segments. Red areas denote the crosstalk violations between two wire segments. After migrating some wire segments to the new layer, no crosstalk violations exist in the new solution as illustrated in Figure 7 (b). Finally, a linear programming approach is presented as post processing to adjust segment positions when the two layers have layer-dependent design spacing rules

6. References

[1] K. Aoshima and M. Iri. Comments on F. Hadlock's paper: finding a maximum cut of a planar graph in polynomial time, SIAM J. Comput., pp. 86-87, 6(1977).

[2] K. Chao. private communication, CPU development, Desktop Platform Business Group, Intel Corporation.

[3] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time, SIAM J. Computing, Vol. 4, No. 3, pp. 221-225, Sept. 1975.

[4] Inquirer. Intel adds two layers to .13 Extreme Editions, http://www.theinquirer.net/?article=14702, Mar. 13, 2004.

[5] N. Karmarkar. A new polynomial-time algorithm for linear programming. Combinatorica 4, 373-395, 1984.

[6] R. M. Karp. Reducibility among combinatorial problems, Complexity of Computation, R.E. Miller and J. W. Thather eds., pp. 85-103, 1972.

[7] L. Khachiyan. A Polynomial algorithm in linear programming, Soviet Mathematics Doklady 20, pp. 191-194, 1979.

[8] Y. S. Kuo, T. C. Chern and W.K. Shih. Fast algorithm for optimal layer assignment, 25th ACM/IEEE Design Automation Conference, pp. 554-559, Anaheim, CA, 1998.

[9] W.-S. Lee, L.-H. Lee, J.-K. Park, T.-K. Kim, Y.-K. Park, and J.-T. Kong. Investigation of the capacitance deviation due to metal-fills and the effective interconnect geometry modeling, Proc. Of 4th Int'l Symp. On Quality Electronic Design (IEEE ISQDE03), pp. 373-376, 2003.

[10] S. Poljak and Z. Tuza. Maximum cuts and large bipartite subgraphs, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, W.Cook, L. Lovasz and P. Seymour eds., pp. 181-244, (20)1995.

[11] T. Sakurai and K. Tamaru. Simple formulas for two and three dimensional capacitance, IEEE Trans. Electron Devices, 1993.

[12] D. D. Sherlekar. Design considerations for regular fabrics, International Symposium on Physical Design, pp.97-102, Apr. 2004.

[13] A. L. Shimpi. AMD Athlon XP 2600/2400+ - Introducing Thoroughbred Revision B, AnandTech, http://www.anandtech.com/cpu/showdoc.html?i=1685&p=3, Aug, 21, 2002.

[14] R. Tian, D. F. Wong, and R. Boone. "Model-Based Dummy Feature Placement for Oxide Chemical-Mechanical Polishing Manufacturability," IEEE Transactions on Computer-Aided Design, Vol. 20, No. 7, 2001.

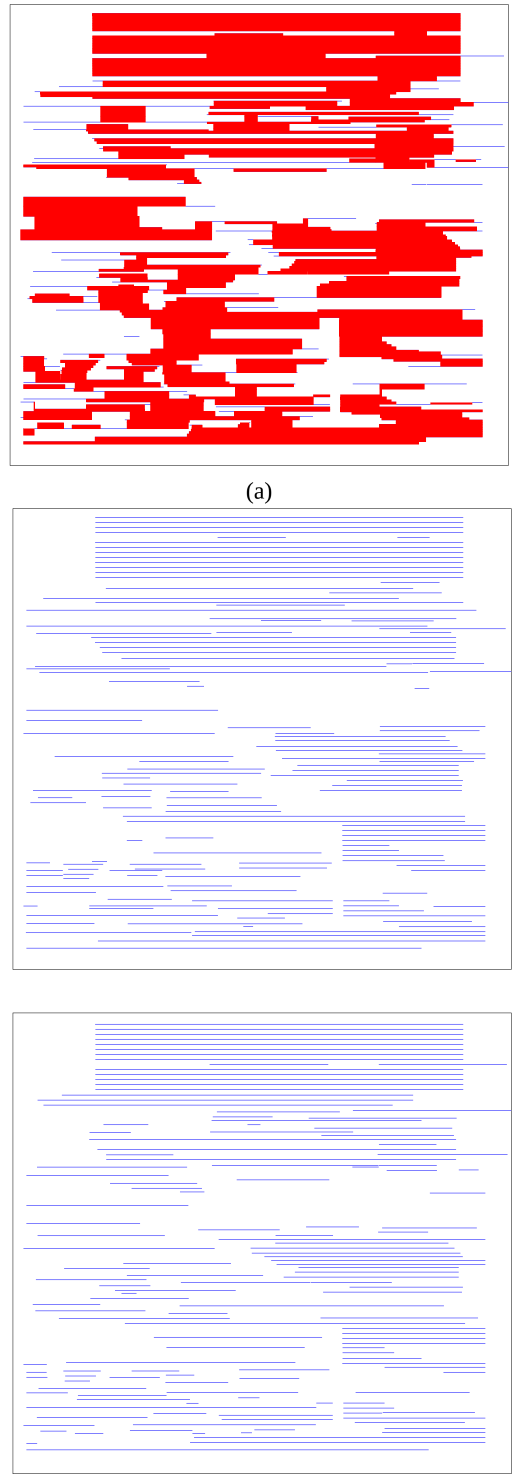
[15] R. Tian, X. Tang, and D. F. Wong. "Dummy Feature Placement for Chemical-Mechanical Polishing Uniformity in a Shallow Trench Isolation Process," IEEE Transactions on Computer-Aided Design, Vol. 21, No. 1, 63-71, 2002.

[16] C. Weber. Yield learning and profitability in the semiconductor industry. http://www.etm.pdx.edu/lecture_series.asp, Apr. 2004.

[17] F. Weber. CTO, AMD, towards instruction set consolidation, presentation in Microprocessor Forum 2003.

[18] R. Wilson. Chip industry tackles escalating mask costs, EETimes, http://www.eetimes.com/semi/news/OEG20020617S0050, June 17, 2002.

[19] H. Zhou and D. F. Wong. Global routing with crosstalk constraints, ACM/IEEE Design Automation Conference, CA, 1998.



(b)

Figure 7: (a) A test file with 281 wire segments. Red areas denote the crosstalk violations between two wire segments. (b) Some wire segments migrate to the new layer, and no crosstalk violations exist in the new solution.