Layer Assignment for High-Performance Multi-Chip Modules

Kai-Yuan Chao

Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, Texas, 78712

D. F. Wong

Department of Computer Sciences
University of Texas at Austin
Austin, Texas, 78712

Abstract

In this paper, we present a layer assignment method for high-performance multi-chip module environments. In contrast with treating global routing and layer assignment separately, our method assigns nets to layers while considering preferable global routing topologies simultaneously. We take transmission line effects into account to avoid noise in high-speed circuit packages. The problem is formulated as a quadratic Boolean programming problem and an algorithm is presented to solve the problem after linearization. Our method is applied to a set of benchmark circuits to demonstrate the effectiveness.

1 Introduction

With the rapid increase in IC performance and system complexity, the packaging delay is becoming a dominant part of total system delay and is expected to be 80% by the year of 2000 [1]. The Multi-chip Module (MCM), which has been developed to eliminate one level of interconnection, places chips on a high density substrate, and therefore decreases the packaging delay drastically. In order to connect a large number of chips (e.g. 131 chips on the IBM3081 TCM) on one carrier, the substrate density for routing may be high (e.g. 8-25 um pitch on MCM-Ds) and the number of wiring layers may be numerous (e.g. 63 layers on the IBM ES9000 TCM). Due to the high wiring density and very fast components contained in high-performance MCM designs, transmission line phenomena which affect circuit performance and signal integrity are getting more attention. Accordingly, it is essential to consider routability, performance, and electrical noise for MCM designs during layer assignment and global routing stages.

In general, two approaches, constrained and unconstrained layer assignments, are used as one phase of the multilayer substrate routing. In constrained layer assignment, the global routing paths are given so the remaining task is to assign wire segments to different layers and to minimize the desired cost such as the number of vias. The second approach assigns nets to different layers first and then routes the nets on each plane. Both problems are NP-complete and have been extensively studied for PCB and IC environments [7]. None of the PCB/IC layer assignment algorithms has considered electrical interference (e.g. cross talk or reflection) that is very likely encountered in MCM designs. A global routing algorithm is presented by [5] to minimize the number of routing layers on MCMs. Recently, [11] proposed a constrained detailed layer assignment method to minimize number of layers and vias. In [4], an unconstrained layer assignment scheme using a bounding box measure of electrical interference and routability is presented for MCM routing. However, both of these approaches consider global routing topologies and layer assignment separately while neglecting the interaction between the two issues. Moreover, no experimental results regarding electrical interference have been reported in previous MCM layer assignment studies. In this paper, we present an unconstrained layer assignment method that considers electrical interference, performance, routability, and global routing topologies for MCM multilayer routing.

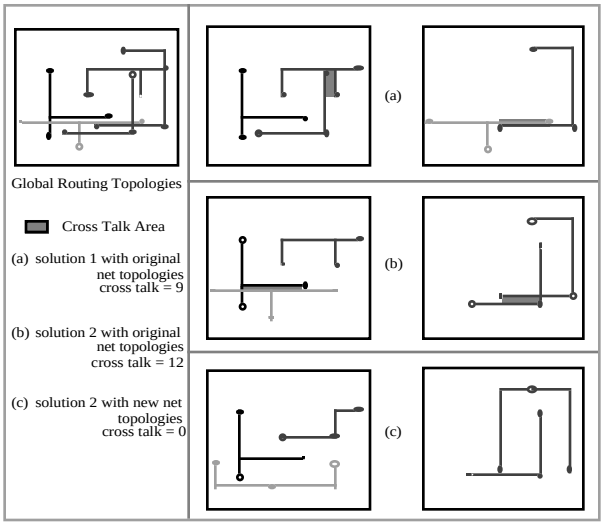

Fig. 1 Interaction between Layer Assignment and Net Topologies (the shaded part between nets is coupling area)

Fig. 1 illustrates the importance for considering global routing topologies during layer assignment stage, where the cost is a function of cross talk noise and number of intersections. Fig. 1(a) shows a solution of layer assignment using the given global routing net

topologies. The solution in Fig. 1(a) is better (actually, optimal) than the solution in Fig. 1(b) if only original topologies are considered. However, the same solution in Fig. 1(b) can be better than the one in Fig. 1(a) if other net topologies are considered (shown in Fig. 1(c)).

For high-performance MCM routing, intersections of wires can cause the use of detour and vias which in turn require more routing resources, lower manufacturing yield, and cause noise problem. Excessive local congestion not only gives rise to future routing difficulty but also increases the potential cross talk noise for high-speed signal lines. Transmission line effects become significant in high-speed electronics if the signal rise time is less than 2.5 times of the propagation time on wire [1]. In MCMs, the average wire length is much longer than the average on-chip wire length. Moreover, to achieve the fastest performance, nearly all contemporary supercomputers and mainframes use high speed bipolar chips on MCMs [6]. For bipolar circuits with an off-chip rise time of 200-400ps on an MCM (with 10cm dimension), it is necessary to consider the transmission line phenomena during global routing and layer assignment.

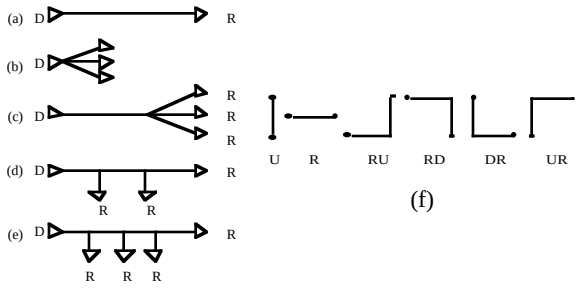

Fig. 2 (a) single-load point to point; (b) multiple-load near-end cluster; (c) far-end cluster; (d) discretely loaded; (e) distributed (same as (d) but with length between receivers $\leq 0.5T_r$); (f) geometrical patterns for two-terminal nets

Due to the discontinuities on transmission lines such as vias and receivers, reflections of signals can occur and increase delay in synchronous signal lines (because of the longer settling time) as well as cause logic failures (because of over/undershoots) [1]. In order to avoid reflections, the line length between each output and input gates should not exceed the critical length which is determined by different device technologies [12]. A multi-terminal net may have many possible routing topologies. However, only several net topologies such as distributed and clustering configurations (see Fig. 2(a)-(e) for an example) are allowable while considering transmission effects [6]. Consequently, the simple bounding box measure used in [4] for layer assignment without taking net topologies into account is not sufficient.

Cross talk is a result of mutual capacitance and inductance coupling between signal lines in close proximity. Cross talk becomes more severe when the lines are closer, the distance from grounding plane is larger, the coupling length is longer, and the rise time is faster [1]. Cross talk not only increases delay (because of the larger effective line capacitance) but also degrades signal integrity and causes logic faults. Reduction of cross talk should be considered for high-performance MCM multilayer routing since the wiring density is high and coupling length between lines may be long.

2 Problem Formulation

A multi-terminal net can be routed in several preferable topologies for avoiding transmission line effects. Instead of impractically processing all configurations of a large fan-out net, we only consider topologies for limited number of nodes and cluster the nearest neighboring nodes as a supernode by pre-connecting them together. A number of topologies are considered among distant supernodes/nodes since the long connecting wire segments contribute to the great majority of coupling, reflection, and delay. The net topologies that have driver-receiver path length longer than the maximum allowable transmission line (reflection) length are not considered. We also do not consider net topologies that cause delay larger than the performance requirement. Each net topology has to be mapped on the routing plane according to geometrical constraints. For traditional rectangular global routing [5], each two-terminal connection of a multi-terminal net can be mapped onto one of the L-shaped configurations shown in Fig. 2(f). Here, we only consider R, U, RU, and RD patterns since only one from {RU, UR} and one from {RD, DR} are sufficient to map a two-terminal wire segment. However, the other three mapping combinations can be used in a post-processing phase to further optimize the result from layer assignment. After geometrical mapping, for two nets on the same layer, the number of wire intersections which may use potential vias in future detailed routing and the wire congestion (or wire density in the global routing tile) can be measured. In order to obtain good routability, the nets should be assigned to appropriate layers such that the number of intersections and local congestion are minimized.

For every pair of nets on the same layer, the cross talk between the nets can be defined by their coupling area. As in [12], we use the *cross talk level* to measure the coupling noise (cross talk) using the following equation.

$$T_x(k,i,p,j,q) = \sum_{\substack{t \in \text{net } i \text{ with topology } p \\ s \in \text{net } j \text{ with topology } q}} \frac{lx_{i,t,j,s}}{lx_{max,k}\, f(dx_{i,t,j,s})}$$

where $T_x(k,i,p,j,q)$ is the x-direction cross talk level between net i with topology p and net j with topology q on layer k, $lx_{max,k}$ is the maximal admissible coupling length for the minimum distance on layer k, $lx_{i,t,j,s}$ is the coupling length between a horizontal wire segment s of net j to a horizontal wire segment t of net i, $dx_{i,t,j,s}$ is

the corresponding coupling distance, and f is a technology-dependent function of distance. The y-direction cross talk level, $T_y(k,i,p,j,q)$, can be calculated in a similar way, and $T_{x+y}(k,i,p,j,q)$ is the total cross talk level.

The interconnection delay of a net can be approximated by the following equation:

$$d_{k,i,p} = \max_{\forall r, r \in \text{net } i} ((l_r r_k + k_d)(l_r + C_r))$$

where $d_{k,i,p}$ is the delay of net i with topology p on layer k, k_d is a coefficient depending on the technology of driver d, l_r is the path length from the driver d to a receiver r, r_k is the corresponding unit wire resistance on layer k, c_k is the unit wire capacitance on layer k, and C_r is the input capacitance of a receiver r. For high-speed transmission lines, regularly, a more sophisticated model than the above equation is required and hence cannot be calculated efficiently even for mid-size circuits. However, it is shown that the performance driven routing prefers cluster topologies to distributed topologies [9]. Hence, bias weights toward different topologies can be added to compensate for the error made from the lumped RC approximation.

We define the interference as the summation of the cross talk level (for both x and y directions) and the number of wire segment intersections. Some group of nets, such as signal wires in analog and digital mixed-signal designs, need to be placed on different layers for avoiding interaction [4]. It is also desired to route clock, power and ground lines on dedicated layers. Hence, as follows, we can formally define sets R_1, R_2, and R_3 to model the constraints mentioned above.

R_1={(i,j)|net i and net j cannot be assigned to the same layer}
R_2={(i,j)|net i and net j must be assigned to the same layer}
R_3={(i,k)|net i must be assigned to layer k }

Note that R_1 and R_2 are mutually exclusive. We define the cost for net i with topology p on layer k and net j with topology q on layer l to be

$$w_{kiplq} = \alpha_1 A_{kiplq} + \alpha_2 T_{kiplq} + \alpha_3 (d_{kip} + d_{ljq})$$

where A_{kiplq} is the total number of intersections between these two nets, $T_{kiplq} = T_{x+y}(k,i,p,j,q)$ if $k=l$ and $T_{kiplq} = 0$ if $k \neq l$, d_{kip} and d_{ljq} are the delays, and α_1, α_2, together with α_3 are constant weights assigned by the designer. Here, we restrict the cost to be zero if $k \neq l$ by assuming that there is no vertical intersection or cross talk (grounding layer in-between). Let Q_i be the number of preferable topologies of net i. Given N nets and K layers, the *layer assignment problem* (LAP) is to assign each net i, $1 \le i \le N$, to its corresponding layer k_i and net topology p_i, where $1 \le k_i \le K$, $1 \le p_i \le Q_i$, such that the total cost is minimized. One special case of this problem where each net has only one topology can be formulated as a max-cut K-coloring problem in which finding a max-cut is NP-complete [4]. In order to consider several preferable topologies for every net, we formulate LAP as a *quadratic Boolean programming problem* (QBP). We define $x_{kip} \in \{0,1\}$, where $x_{kip}=1$ if net i with topology p is assigned to layer k, and $x_{liq}=0$, $\forall l \neq k$, $1 \le l \le K$, $\forall q \neq p$, $1 \le q \le Q_i$. Accordingly, we are trying to find a 0-1 matrix $\mathbf{X}=[x_{kip}]$ that minimizes

$$\sum_{k_1=1}^{K} \sum_{i_1=1}^{N} \sum_{p_1=1}^{Q_{i_1}} \sum_{k_2=1}^{K} \sum_{i_2=1}^{N} \sum_{p_2=1}^{Q_{i_2}} x_{k_1 i_1 p_1} w_{k_1 i_1 p_1 k_2 i_2 p_2} x_{k_2 i_2 p_2} \qquad (1)$$

subject to the following constraints:

$$C_0 \ : \ \sum_{k=1}^{K} \sum_{p=1}^{Q_i} x_{kip} = 1, \ \forall i, \ 1 \le i \le N$$

$$C_1 \ : \ \sum_{p=1}^{Q_i} x_{kip} + \sum_{q=1}^{Q_j} x_{kjq} < 2, \ \forall (i,j) \in R_1, \ \forall k, \ 1 \le k \le K$$

$$C_2 \ : \ \sum_{p=1}^{Q_i} x_{kip} = \sum_{q=1}^{Q_j} x_{kjq}, \ \forall (i,j) \in R_2, \ \forall k, \ 1 \le k \le K$$

$$C_3 \ : \ \sum_{p=1}^{Q_i} x_{kip} = 1, \ \forall (i,k) \in R_3$$

where C_0 is the *assignment constraint* which states that each net can only be assigned to one layer and to one topology. The *conflict constraint* C_1, *common-layer constraint* C_2, and *pre-assignment constraint* C_3 are specified by R_1, R_2, and R_3, respectively. We will address constraints C_2 and C_3 in the Section 3.3 and will consider constraints C_0 and C_1 hereafter.

3 Layer Assignment Algorithm

We transform the three-dimensional matrix $\mathbf{X}=[x_{kip}]$ in (1) into a one-dimensional vector $\mathbf{y}=[y_j]$ of size KP by defining $y_j=x_{kip}$ and $j=kP+\Delta_i+p$, where $P = \sum_{i=1}^{N} Q_i$ and $\Delta_i = \sum_{j=1}^{i-1} Q_j$. This transformation is a one-to-one mapping since it is the concatenation of elements in a three-dimensional matrix. Similarly, for each $w_{k_1 i_1 p_1 k_2 i_2 p_2}$, we define its corresponding $m_{j_1 j_2}$, where $j_1=k_1P+\Delta_{i_1}+p_1$, $j_2=k_2P+\Delta_{i_2}+p_2$, and $\mathbf{M}=[m_{j_1 j_2}]$ is a KPxKP non-negative cost matrix. The constraints are also transformed in the same way. Accordingly, the problem is rewritten as

$$\text{minimize } \mathbf{y}^{\mathrm{T}}\mathbf{M}\mathbf{y}, \text{ subject to } C_0 \text{ and } C_1 \qquad (2)$$

3.1 Quadratic Boolean Programming

If we define the solution space to be $S=\{\mathbf{u} | \ \mathbf{u}$ is a vector satisfying C_0 and $C_1\}$, then we can find a constant vector $\mathbf{g}=[g_j]$ satisfying the following condition since both N and each $w_{k_1 i_1 p_1 k_2 i_2 p_2}$ are bounded.

$$g_j \ge \sum_{i=1}^{KP} m_{ji} y_i, \ \forall \mathbf{y} \in S, \ \forall 1 \le j \le KP \qquad (3)$$

Furthermore, a corresponding vector $\mathbf{a}^{(k)}=[a_j]^{(k)}$ and a number $b^{(k)}$ for each feasible solution $\mathbf{u}^{(k)}=[u_j]^{(k)}$ (at iteration k) are defined as follows.

$$a_j^{(k)} = \sum_{i=1}^{KP} m_{ij} u_i^{(k)} + g_j u_j^{(k)}, \quad b^{(k)} = \sum_{i=1}^{KP} g_i u_i^{(k)} \tag{4}$$

Proved in [2, 3], every optimal solution $\mathbf{y}^*$ of expression (2) corresponds uniquely to an optimal solution $(z^*, \mathbf{y}^*)$ of

$$\min_{\mathbf{y}\in S} z \text{, such that } z \ge \sum_{j=1}^{KP} a_j^{(k)} y_j - b^{(k)}, \text{ or}$$
$$\min_{\mathbf{y}\in S} z \text{, such that } z \ge \max_{1\le j\le KP} a_j^{(k)} y_j - b^{(k)} \tag{5}$$

Since directly solving linearlized expression (5) would require tremendous amount of storage and time, [3] proposed the following heuristic.

Burkard's Heuristic

1. Initialize k=1, $h_j^{(0)}$=0, $\forall 1\le j\le KP$
2. Compute bounds g_j, $\forall 1\le j\le KP$, and start with a feasible $\mathbf{u}^{(1)}\in S$ with setting $\mathbf{u}^*=\mathbf{u}$, $z^*=\mathbf{u}^{*\mathrm{T}}\mathbf{M}\mathbf{u}^*$
3. Compute $a_j^{(k)} = \sum_{i=1}^{KP} m_{ij} u_i^{(k)}, \forall 1\le j\le KP$, and $b^{(k)} = \sum_{i=1}^{KP} g_i u_i^{(k)}$
4. Solve $z = \min_{\mathbf{u}\in S}(\sum_{j=1}^{KP} a_j^{(k)} u_j)$ (here $\mathbf{u}$ is a variable vector)
5. Compute $h_j^{(k)} = h_j^{(k-1)} + \dfrac{a_j^{(k)}}{\max(1, |z - b^{(k)}|)}, \forall 1\le j\le KP$
6. Solve $\min_{\mathbf{u}\in S}(\sum_{j=1}^{KP} h_j^{(k)} u_j)$ and let $\mathbf{u}^{(k+1)}$ be the solution
7. If $(\mathbf{u}^{(k+1)})^{\mathrm{T}}\mathbf{M}\mathbf{u}^{(k+1)} < z^*$ then $\mathbf{u}^*=\mathbf{u}^{(k+1)}$ and $z^*=(\mathbf{u}^{(k+1)})^{\mathrm{T}}\mathbf{M}\mathbf{u}^{(k+1)}$
8. if $k \le N_{iteration}$ then $k=k+1$ and go to 3. Otherwise stop

The well known Burkard's heuristic has been applied to solve large size quadratic assignment problems [3] (special forms of QBPs) and VLSI system partition problems [10]. This heuristic can find a good sub-optimal solution and has relatively stable performance with respect to the initial solution and bounding vector $\mathbf{g}$ [3]. We use the parameter, $N_{iteration}$, to control the running time and the quality of final solution.

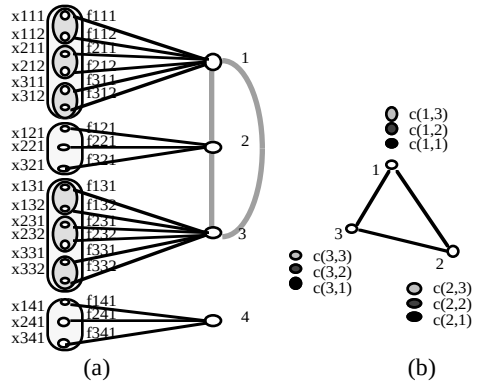

Fig. 3 (a) GAP (shaded lines are conflict constraints); (b) WCP (the conflict graph transformed from (a))

3.2 Maximum Penalty Coloring

In Burkard's heuristic, steps 4 and 6 are, in fact, two special *general assignment problems* (GAP) which is to assign a net i to the layer k with topology p, $1\le i\le N$, such that the summation of corresponding coefficient f_j (which is one-to-one mapped to f_{kip} while $f_j=a_j$ at step 4 or $f_j=h_j$ at step 6) is minimized. In Fig. 3(a), the example has 4 nets to be assigned to 3 layers and nets 1 and 3 have two topologies. The GAP for this example is to choose one f_{kip} (and assign the corresponding x_{kip}=1) for each net i, $1\le i\le 4$. If there is no constraint C_1, the problem becomes to find, for every net i, the minimum coefficient f_j from corresponding coefficients of x_{kip}, $1\le k\le K$ and $1\le p\le Q_i$, which can be solved efficiently. For net 4 in Fig. 3(a) constrained by C_0 only, we assign $\min\{f_{141}, f_{241}, f_{341}\}$ to net 4. The assignment constraint C_0 forces each set of binary variable X_i, where $X_i=\{x_{kip}|1\le k\le K, 1\le p\le Q_i\}$, can only have one element to be one. Therefore, in the corresponding coefficient set, $F_i=\{f_{kip}|1\le k\le K, 1\le p\le Q_i\}$, equivalently, only one coefficient is chosen to contribute to z, the inner product sum ($\mathbf{a}\bullet\mathbf{u}$ at step 4 and $\mathbf{h}\bullet\mathbf{u}$ at step 6). In order to minimize z, we have to choose the minimum coefficient in each F_i. Accordingly, we have the following theorem.

Theorem *If there is no conflict constraint C_1, two GAPs in step 4 and step 6 can be solved optimally in O(KP) time, where $P = \sum_{i=1}^{N} Q_i$. However, to solve the two GAPs in step 4 and step 6 under constraint C_1 is NP-hard.*

We define I to be a set of N nets. Let I_c be the set of nets constrained by conflict constraint C_1 and let I_{nc} be the set of nets without restriction from C_1. Note that $I = I_{nc} \cup I_c, I_{nc} \cap I_c = \varnothing$. Since the nets in I_{nc} can be solved efficiently and independently in both GAPs (but neither in the original LAP nor QBP), the remaining task is to solve the constrained nets in I_c. However, due to the constraint C_1, it is more difficult to solve both GAPs. First, to determine if there is a feasible solution under conflict constraint is difficult, which can be proved by transforming both GAPs to the well known graph K-colorability problem. Moreover, finding the solution to minimize the GAP in step 4 or step 6 is also NP-hard even feasible solutions are available. In real applications, there should be enough layers to accommodate those conflicting nets in R_1 (otherwise, layer number should be increased). Henceforth, we restrict the maximum number of conflicts each net can have in R_1 is less than K such that a feasible solution is guaranteed. In order to solve the problems, both GAPs under C_1 is transformed to the *weighted vertex coloring problem* (WCP and see Fig. 3(b) for an example) which is to minimize the coloring cost of a K-colorable graph and, there is a weight for each vertex to be assigned to a certain color. More precisely, we construct a *conflict graph* $G=(V,E)$. $\forall (i,j)\in R_2 \Leftrightarrow \exists (v_i,v_j) \in E$, $|V|=|I_c|$, $|E|=|R_1|$, and let

$c(v_i,k)$ be the coloring cost of vertex v_i assigned to color k, $\forall v_i \in V$, $1 \le k \le K$, where $c(v_i,k)=f_{ki,min}=\min\{f_{kip}, 1 \le p \le Q_i\}$. The objective is to find an assignment $f: V \rightarrow \{1, 2, ..., K\}$ such that $f(v_i) \neq f(v_j)$ whenever $(v_i,v_j) \in E$ and minimize $\sum_{i=1}^{|V|} c(v_i, f(v_i))$. In Fig. 3(b), the conflict graph is constructed from net 1, net 2, and net 3 constrained by C_1 of the GAP in Fig. 3(a). For example, $c(1,k)=\min\{f_{k11}, f_{k12}\}$, $1 \le k \le 3$, for net 1 in Fig. 7(a). The minimization of coloring cost in WCP is equivalent to the minimization of summation in the original GAP. Accordingly, we propose a *maximum penalty coloring* (MPC) algorithm to solve WCP transformed from the GAP under conflict constraint C_1. Here, the penalty for each vertex is the difference of its secondary minimum coloring cost (a large constant if only one color is allowable) and its minimum coloring cost.

Algorithm MPC

1. sort coloring costs and calculate penalty for every $v \in V$,
2. choose a vertex v with maximum penalty. if more than one vertex have the same penalty, choose the one with least neighboring vertices (vertices that connects v) that are not colored yet.
3. color v with its current minimum-cost color c.
4. delete c from the allowable colors of all the neighboring vertices of v and update their penalties.
5. $V=V \backslash v$ and go to Step 2 if $|V| > 0$.

This $O(K|V|\log K+|V|^2)$ MPC algorithm first colors the node with maximum penalty (if it does not get the assigned color at this iteration) and with the least degree of freedom in choosing colors. The original GAP in step 4 or step 6 under constraint C_1 can therefore find a suitable solution in $O(K N_c(\log K + Q) + N_c^2)$ time (including the transformation to WCP) where Q is the maximum topology a net can have and $N_c=|I_c|$.

3.3 Extensions

Other Constraints: The common-layer constraint C_2 can be added by assigning nets i and j, $\forall (i,j) \in R_2$, to the same color in the MPC algorithm. While processing I_{nc} or I_c, we treat (i,j) as a combined supernode $i+j$ and use the corresponding $f_{k,i+j,(p,q)}=f_{kip}+f_{kjq}$ as the new coefficient. If the total number of combined topologies of the supernode is large, we only consider limited number of most preferable combined topologies and then post process nets in the supernode (for choosing best topologies on the same layer) after k is determined. For constraint C_3, we pre-assign net i to layer k, $\forall (i,k) \in R_3$, and only process coefficient f_{kip}, $1 \le p \le Q_i$.

Vertical Cross Talk: If there is a ground routing plane in-between each pair of signal routing plane (or x-y plane pair), then the vertical cross talk noise is negligible. For MCM designs without grounding planes in-between signal planes either because of technical or economical reasons, we have to consider vertical cross talk between planes. Hence, we re-define the cost matrix: $\forall k_1 \neq k_2$, $w_{k_1 i_1 p_1 k_2 i_2 p_2} = 0$ if vertical interference is not considered, or $w_{k_1 i_1 p_1 k_2 i_2 p_2} = F(k_1,k_2)(T_{x+y}(k_1,i_1,p_1,i_2,p_2)+T_{x+y}(k_2,i_1,p_1,i_2,p_2))$ if vertical interference is considered, where $F(k_1,k_2)$ is the a constant coefficient depending on technologies and vertical distance between layer k_1 and layer k_2 in MCM. In contrast to [4], our method can optimize intra-plane and inter-plane interference at the same time without permutating layers after layer assignment.

Circuit Partitioning: Large MCMs may have more than thousands of nets. The storage used for cost matrix **M** may be large. Note that the storage requirement of **M** is $O(P)$ in our method and independent of the number of layers since each cost in **M** is derived from the single plane cost, where P depends on the number of nets, N, and the maximum preferable topologies each net can have. Therefore, we propose a method to partition large circuits such that the problem can be handled under allowable resource requirements. In Fig. 4, we process the center part (shaded area) first, and then deal the corner parts later by fixing the partial solution we derived from the previous process. Similarly, with increasing the number of overlapping center regions, we can partition the problem into more than 5 smaller subproblems for even huge circuits. Experimental results in the following section show that this partitioning method can achieve comparable solutions and save both storage and computation resources.

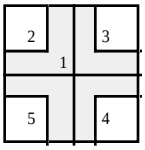

Fig. 4 Partitioning of Circuits

4 Experimental Results and Conclusions

The C programs are tested on a Sun IPX with 32MB memory. We apply our method on five benchmark circuits in Table 1 where we assume MCM-C and MCM-D technologies are used. The mcm213 and mcm848 are from [11]. The mcc1, mcc2-75, and mcc2-45 circuits are from MCNC. The large mcc2 circuit with 7118 nets is actually a supercomputer core using 37 VHSIC gate arrays. Without loss of generality, we assume that drivers and receivers use BCT technology [12] and derive the corresponding factors from dielectric constants of MCM-C (5.5) and MCM-D (4.0). Unfortunately, there is no driver/receiver specification in these benchmark circuits. Therefore, we use wirelength to measure the performance and spanning tree configurations for simplicity. In this experiment, we assign that $\alpha_1=0.05$, $\alpha_2=4.0$, and $\alpha_3=0.4$, and let the maximum number of topologies per net be 16.

Table 2 shows the result for different number of layers used for assignment where X Talk is total cross talk level, and congestion is the exceeding number of wire segments in global routing tile regarding to the number of tracks. The routability/planarity is increased, and cross talk is decreased when more layers are used for routing. The solution quality and program running time depends on the number of iteration. We present the total cost for different numbers of iterations in Table 3 where the number of assigned layers we use hereafter is the lower bounds (in Table 4) from [11] and from a detailed router of [8]. As shown in the table, the more iterations it takes, the better solution it gets. The running time also includes the data I/O time, and therefore, it takes longer to preprocess cost matrix for larger input circuits. Since the reduction rates of cost for iterations after 20 are not high in our experiment, we henceforth run 20 iterations for testing the benchmark circuits. For comparison, we also implement the max-cut K-coloring algorithm (MC) [4] to assign layers (by using the single topology weight which is derived from the average weights for different topologies) and then use our method (by letting K=1) to choose the best global routing topologies for nets on each layer. The results for the large circuit mcc2 are derived by partitioning into 5 subcircuits. From the results shown in Table 4 (where QBP* is the partitioning version of QBP), our method perform better on average of 37.8% of cost reduction, 29.6% of cross talk decrease, 43% of intersection reduction, and 85.2% of less congestion at the charge of longer computation time. For circuits with more multi-terminal nets (and thus more topologies), our results are much better than those achieved by using the MC method that considers layer assignment and global routing topologies separately. Even for the mcc2 circuit that contains 95% of 2-terminal net which has one topology, our method still shows better outcomes. In Table 5, we show the results with and without partitioning. It is indicated that our partitioning method can get comparable results (with increasing average 10% in cost) while saving average 40% running time and 70% storage in our experiment.

We propose a method to optimize routability, performance, and electrical noise interference for high-performance MCM layer assignment. Instead of treating global routing and layer assignment separately, our method considers different preferable global routing topologies and layer assignment at the same time. Experimental results show that our method achieves lower level of cross talk, less local congestion than processing layer assignment without considering global routing topologies.

Acknowledgments

This work was partially supported by the Texas Advanced Research Program under Grant No. 003658459. The authors thank Dr. Charles W.-C. Lin and Dr. Patrick Jaillet for valuable discussions.

References

[1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publ. Co., 1990.

[2] E. Balas and J. B. Mazzola, "Quadratic 0-1 Programming by a New Linearization," Presented at the ORSA/TIMS National Meeting, Washington, D. C., 1980.

[3] R. E. Burkard and T. Bonniger, "A Heuristic for Quadratic Boolean Programs with Applications to Quadratic Assignment Problems," Euro. J. of Operational Res., v. 13, pp. 374-386, 1983.

[4] J. D. Cho, S. Raje, M. Sarrafzadeh, and et. al., "Crosstalk-Minimum Layer Assignment," Proc. of IEEE Custom Integrated Circuits Conf., pp. 29.7.1-29.7.4, 1993.

[5] J. M. Ho, M. Sarrafzadeh, G. Vijayan and C. K. Wong, "Layer Assignment for Multichip Modules," IEEE trans. on CAD, v. 9, no. 12, pp. 1272-1277, 1990.

[6] E. E. Davidson, P. W. Hardin, et. al., "Physical and Electrical Design Features of IBM Enterprise System/9000 Circuit Modules," IBM J. Res. Develop., v. 36, no. 5, 1992.

[7] D. A. Joy and M. J. Ciesielski, "Layer Assignment for Printed Circuit Boards and Integrated Circuits, " Proc. of the IEEE, v. 80, no. 2, pp. 311-331, 1992.

[8] K.-Y. Khoo and J. Cong, "An Efficient Multilayer MCM Router Based on Four-Via Routing," Proc. of the 30th DAC, pp. 590-595, 1993.

[9] D. P. LaPotin, "Early Assessment of Design, Packaging and Technology Tradeoffs," Int'l J. of High Speed Electronics, v. 2, no. 4., 209-233, 1991.

[10] M. Shih and E. S. Kuh, "Quadratic Boolean Programming for Performance-Driven System Partitioning," Proc. of the 30th DAC, pp. 761-765, 1993.

[11] M. Sriram and S. M. Kang, "Detailed Layer Assignment for MCM Routing," Proc. of ICCAD, pp. 386-389, 1992.

[12] D. Theune, R. Thiele, T. Lengauer, and A. Feldmann, "HERO: Hierarchical EMC-Constrained Routing," Proc. of ICCAD, pp. 468-472, 1992.

Circuits	# Nets	# Pins	Grid/Global Tile	Pitch/Material
mcm213	213	694	80x80/10x10	200um/ceramic
mcm848	842	2751	150x150/10x10	200um/ceramic
mcc1	802	2043	600x600/10x10	75um/thin film
mcc2-75	7118	14661	2033x2033/40x40	75um/thin film
mcc2-45	7118	14661	3387x3387/60x60	45um/thin film

Table 1 Benchmark Circuits

Circuits	mcm213		mcc1	
# Layers	K=4	K=7	K=4	K=7
Cost	661	286	18279	11704
X Talk	33	16	12807	1251
Intersection	1314	552	27559	16720
Congestion	284	26	0	0
CPU(min.)	2.69	3.60	20.58	26.93

Table 2 Results from Different Number of Layers

$N_{iteration}$		0	10	20	30	40
mcm213	Cost	3616	297	286	280	278
	CPU(min.)	1.03	2.32	3.60	4.94	6.24
mcc1	cost	330560	23522	18279	17701	17428
	CPU(min.)	6.25	13.47	20.58	27.67	34.76

Table 3 Results from Different Number of Iterations

Circuits	mcc1		mcc2-75		mcc2-45	
Method	QBP	QBP*	QBP	QBP*	QBP	QBP*
Cost	18279	18880	225377	251575	395058	459202
X Talk	1807	1667	28425	29871	41955	40390
Intersect.	27559	30329	278447	328883	481392	530875
Congest.	0	0	531	408	566	554
CPU(min)	20.58	15.93	415.13	181.72	420.35	189.28

Table 5 Results after Partitioning

Circuits	mcm213 (K=7)		mcm848 (K=10)		mcc1 (K=4)		mcc2-75 (K=6)		mcc2-45 (K=4)	
Method	QBP	MC	QBP	MC	QBP	MC	QBP*	MC	QBP*	MC
Cost	286 (-58%)	686	1225 (-74%)	4720	18279 (-30%)	26164	251575 (-15%)	296859	459202 (-12%)	521227
Wire	10516	10519	57510	57522	372538	372593	5411885	5411887	9018234	9018229
X Talk	16 (-40%)	27	54 (-67%)	165	1807 (-25%)	2381	29871 (-4%)	30990	40390 (-13%)	46446
Inter-section	552 (-61%)	1430	2509 (-75%)	9952	27559 (-34%)	41620	328883 (-23%)	429574	530875 (-22%)	677219
Con-gestion	26 (-85%)	179	0 (-100%)	57	0 (-100%)	10	408 (-62%)	1062	554 (-79%)	2597
CPU (min.)	3.60	1.46	68.38	19.56	20.58	12.04	181.72	96.76	189.28	101.50

Table 4 Comparison of Different Methods