

A New Clustering Method Based on General Connectivity

WENJUN ZHUANG, YONG CHING LIM, and GANESH SAMUDRA

Department of Electrical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 0511, Republic of Singapore.

NENG YAN

Department of ICCAD, Beijing Software Engineering Center, P.O. Box 912, Chinese Academy of Science, Beijing 100083, People's Republic of China.

In this paper, we present a novel concept of the general connectivity among cells. While conventional concept considers direct connections only, the new concept considers both the direct and indirect connections among cells leading to a model capturing a more precise relationship among cells. Based on the model, a new parallel clustering approach is proposed and analyzed. Another new concept of the stable cluster is introduced to improve the clustering result. In conjunction with the clustering process, an investigatory procedure which back tracks the cluster development process to check if all clusters are stable is also developed. Initial test runs, using the new clustering approach for placement, yield an encouraging 21.6% reduction on the number of feed-throughs in a complex ASIC design.

Key Words: *VLSI layout, placement, cluster, parallel clustering, stable cluster, general connectivity.*

1 INTRODUCTION

The placement problem in VLSI layout has attracted a lot of attentions from researchers in the past decades due to its importance in reducing the chip area and improving the chip performance. The problem has been proved to be NP-hard [14]. Therefore, there have been several independent efforts [13, 15] directed towards obtaining the optimal placement solution efficiently.

In recent years, there have been many reports on the application of the simulated annealing method to optimize cell placement in VLSI layout [7–10]. However, for some situations the method based on the simulated annealing not only requires excessive computing resources but also produces results no better than those produced by simple heuristic methods [11, 12]. For example, consider the problem with only 50 cells and 500 nets [11]. The simulated annealing with 3 annealing schedules and 2 allowed moves from 10 different initial placements produced the best result with a wire length of 1887 units in 900 seconds. On the other hand, the simple greedy algorithm [16] achieved the wire length of 1842 units

in just 18 seconds. This is due to the following primary reasons.

(1) The method based on the simulated annealing is effective in cases when (a) the envelop of the convergence curve of the objective (cost) function is monotonically decreasing and (b) the separations between a trough and its subsequent peak of the convergence curve are also monotonically decreasing, as shown in Figure 1. Unfortunately, for an NP-hard problem, this is not generally the case. It is then very difficult to implement an efficient annealing schedule to reach the optimal solution in many situations.

(2) Because of the unpredictable nature of the convergence process, the method based on the simulated annealing degenerates into a random search producing a list of local optimal solutions. The quality of the local optimal solutions improves as the search time increases. This results in an excessive computational time or in a failure to reach an expected solution in a limited time for large circuits.

In most of the existing iterative improvement algorithms, a pair of cells is selected at a time and their locations swapped. In many cases, the swap may not be successful due to the connectivity constraints im-

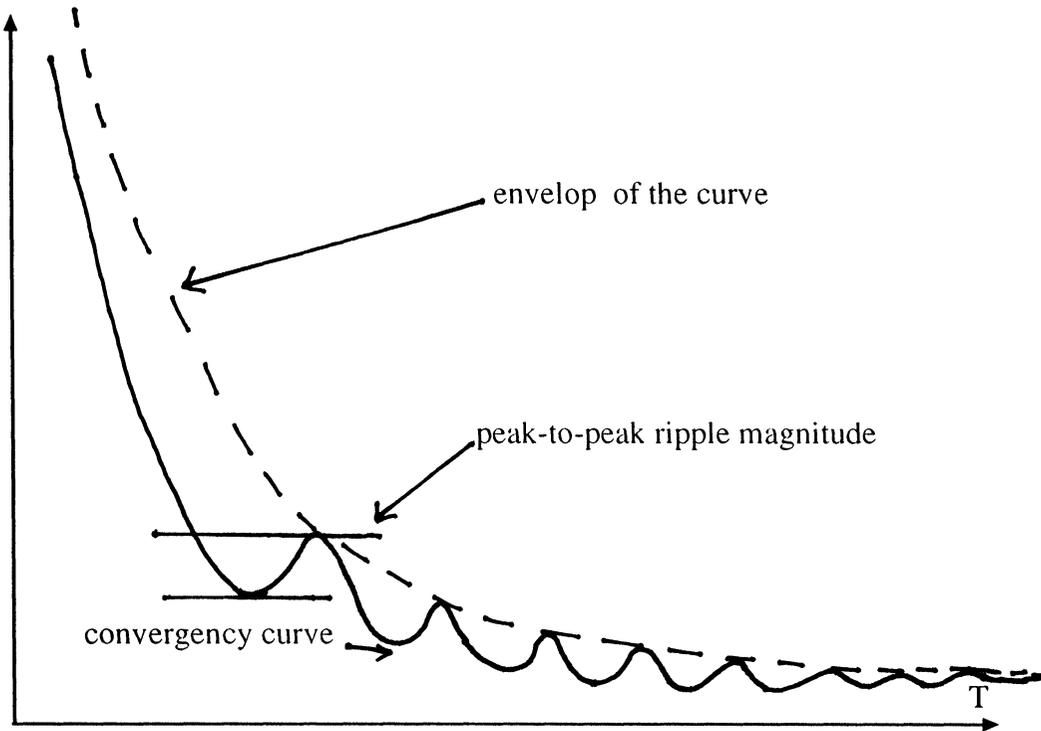


FIGURE 1 Convergency curve of the objective function.

posed by the current topology of the cells already placed. For the example shown in Figure 2, if the distances within the cells on the left and the right are small and the distance d between the left and right is large, then any swap between a single cell on the left and a cell on the right will not be successful. This seriously restricts the effectiveness of all such improvement algorithms including that of the simulated annealing. On the other hand, the swap be-

tween the group (A, B) and the group (H, I, J) is desirable because of improved wire length. However, the group-swapping is very difficult to implement due to the identification of the groups to be swapped and the involvement of groups with different size. Therefore, the development of techniques to avoid such unreasonable topology in the initial placement is very important.

Let us further observe how the initial placement affects its corresponding final placement. Figure 3 shows our experimental results for the number of feed-throughs in the final placement starting from 34 different initial placements. The results are derived from the circuit HK5851 which consists of 1724 modules and 1893 nets. In the experiment, the construction algorithm growing from the opposite sides of the chip is applied for the initial placement. Then the pair exchange algorithm is invoked to arrive at the final placement [6]. The relationship between the number of feed-throughs in the initial placement and that in the final placement is approximately linear. Hence, a better initial placement is more likely to lead to an improved final placement. Similar results was first reported by Goto [1]. Therefore, it is especially important to obtain a good initial placement.

It can also be seen from Figure 3 that several points deviate from the linear relationship. This may be due to the following reasons:

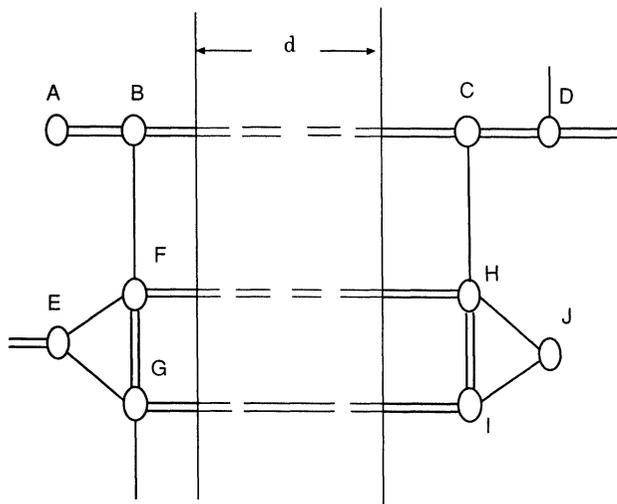


FIGURE 2 A unreasonable topology of cells.

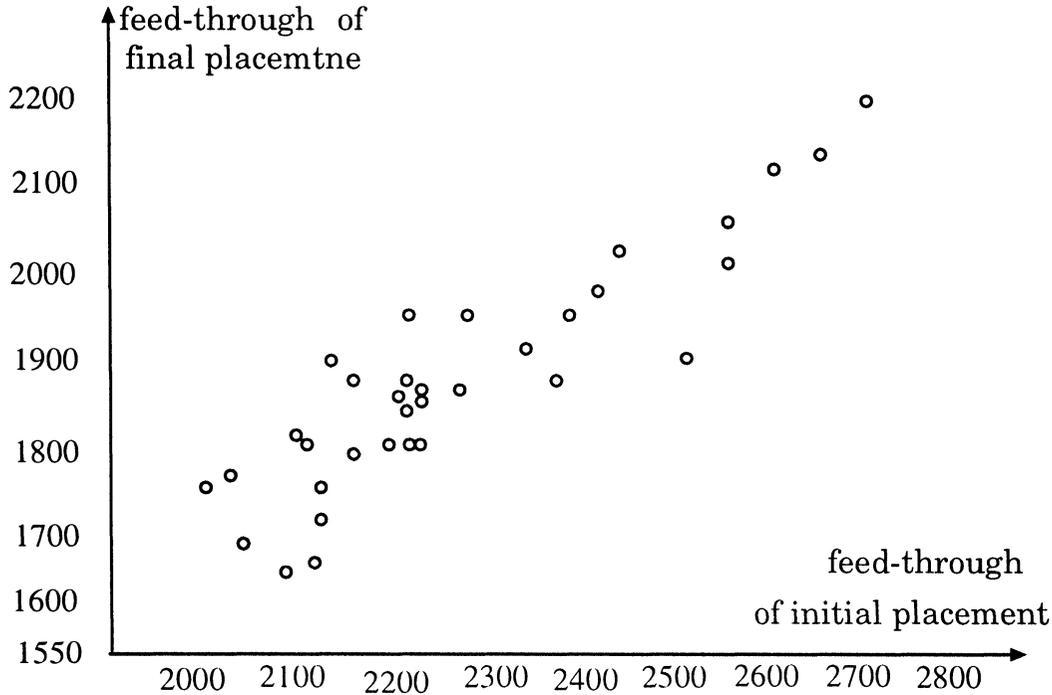


FIGURE 3 34 Experimental results of the circuit HK5851.

(1) In most of the widely used techniques for the construction of initial placement, cells are generally being placed one at a time. The selection of a cell for placement is only based on its connectivity with the other cells. It is impossible to use more accurate criteria such as total wire-length and chip-area because they cannot be evaluated until all the cells are placed. Hence, the quality of initial placement depends critically on the accurate description of the connectivity among cells. However, the traditional description of the connectivity fails to provide sufficiently accurate information for an effective initial placement.

(2) During the iterative improvement phase, a pair of cells is selected to swap their locations. However, the swap may not be successful due to the connectivity constraints imposed by the topology of the cells placed during the initial or intermediate placement (Figure 2).

It is clear from this description that to obtain a satisfactory final placement, it is essential to improve the quality of the initial placement with the help of an accurate description of the connectivity among cells and an avoidance of the unreasonable topologies such as that shown in Figure 2. In this paper, we present a novel concept of the general connectivity among cells and a new parallel clustering algorithm to achieve the above objective.

This paper is organized as follows. Section 2 introduces the concept and evaluation of the general connectivity. Section 3 describes the principle of parallel clustering. Section 4 presents an algorithm for the parallel clustering. Section 5 is devoted to the applications, experimental results and discussions about further study. Section 6 concludes the paper.

2 GENERAL CONNECTIVITY

We introduce a new notion of general connectivity which allows representation of both the direct and indirect connections among cells. It yields a more accurate representation of the cell relationship as compared to traditional approaches which consider only the direct connectivity between cells. To define the k^{th} order general connectivity between cell i and cell j , we introduce a concept of a connection graph and a conductance networks for a given circuit.

Definition 1: ν is a set of vertices in which each vertex represents a cell to be placed.

Definition 2: e is a set of edges. An element e_{ij} is an edge connecting vertices ν_i and ν_j if there is a connection between cell i and cell j . The weight of an edge is a measure of the connection strength.

We denote the connection graph consisting of e and ν by $P(e, \nu)$.

Definition 3: $p(i, j, k)$ is a subgraph of $P(e, v)$ where an edge and its related vertices are elements of the subgraph $p(i, j, k)$ if and only if the edge lies in a path connecting vertices v_i and v_j with the path length no more than k .

Definition 4: $p^*(i, j, k)$ is a conductance network corresponding to the subgraph $p(i, j, k)$, in which each vertex in $p(i, j, k)$ is replaced by an electrical node and each edge in $p(i, j, k)$ is replaced by a conductor with admittance equal to the weight of the edge.

Definition 5: The k^{th} order general connectivity between cell i and cell j , denoted by $G(i, j, k)$, is defined to be numerically equal to the admittance between node i and node j of the conductance network $p^*(i, j, k)$. Obviously, $G(i, j, 1)$ represents the direct connectivity between cell i and cell j and the general connectivity is always greater than zero.

We illustrate the significance of the general connectivity by using two simple examples. The connection subgraph of the first example has a total of 5 vertices and 4 edges with the weights as indicated in Figure 4. The corresponding conductance network is also shown in the figure. Since there is only one conductor between node 1 and node 2, $G(1, 2, 1) = 2q$. Similarly, $G(3, 4, 1) = G(1, 5, 1) = G(5, 2, 1) = 2q$. In order to determine the second order general connectivity between cell 1 and cell 2, we have to evaluate the total conductance between node 1 and node 2 where all the connections with one or two conductances between them are included. Hence, $G(1, 2, 2) = 3q$ since the connection through node 5 is included and $G(3, 4, 2) = 2q$. The traditional approaches (which essentially employ the first order general connectivity) cannot account for the indirect connectivity through node 5. Thus, we have shown by means of the second order general connectivity that the relationship between cell 1 and cell 2 is stronger than that between cell 3 and cell 4.

Let us illustrate the utility of the general connectivity to arrive at a good initial placement in the example. Consider the following situation during the

initial placement. Assume that cell 1 has been placed and cell 2 and cell 5 have not yet been placed. Also assume that a cell will be placed only if its connectivity with the cells already placed is not less than $2.5q$. In this case, neither cell 2 nor cell 5 can be placed based on the first order connectivity. This may result in an unreasonable initial placement topology. However, based on the second order general connectivity, cell 2 will be placed followed by cell 5. In fact, after cell 5 and cell 2 have been placed, the direct connectivity of any cell among cell 1, cell 2 and cell 5 with other placed cells is not less than $4q$ showing a good quality initial placement.

Similar phenomenon is also observed in the clustering process. Although it is obvious that cell 1, cell 2 and cell 5 should be clustered together, based on the traditional direct connectivity approach, any cell-pair among cell 1, cell 2 and cell 5 cannot be clustered because the connectivity of any cell-pair is not strong enough due to the influence of unclustered cells. Such situations will always lead to a major problem for cell clustering. However, it is easy to cluster the cells based on the general connectivity. Hence, the general connectivity is useful in the improvement of the initial placement, the clustering process, and an avoidance of the construction of unreasonable initial placement topology.

The significance of the general connectivity is further illustrated by using another example. Figure 5 shows the subgraph for the determination of the 3rd order general connectivity between cell A and cell B . In the corresponding conductance network, the admittance measured between node A and node B is $2.5q$ Siemens. Hence, the 3rd order general connectivity between cell A and cell B , $G(A, B, 3)$ is $2.5q$. We can explain the general connectivity by using another way;

$$G(A, B, 3) = G(A, B, 1) + [G(A, B, 2) - G(A, B, 1)] + [G(A, B, 3) - G(A, B, 2)],$$

$$G(A, B, 3) = q + q + 0.5q = 2.5q$$

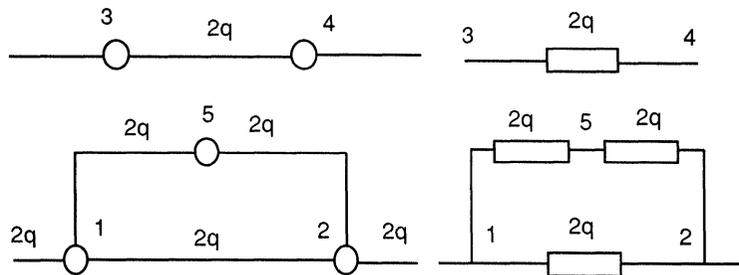


FIGURE 4 A example illustrating the significance of the general connectivity.

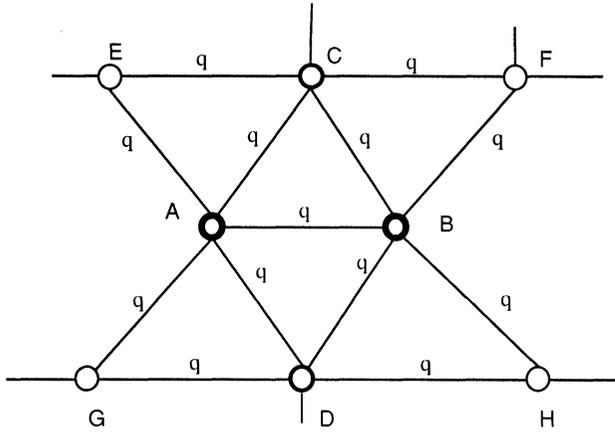


FIGURE 5 A subgraph for the determination of the 3rd order general connectivity.

In Figure 5, it is obvious that $G(A, B, 1)$ corresponds to the direct connection between cell A and cell B , $[G(A, B, 2) - G(A, B, 1)]$ corresponds to the indirect connection of cell A and cell B through cell C and cell D , and $[G(A, B, 3) - G(A, B, 2)]$ corresponds to another indirect connection of cell A and cell B through cell-pairs (E, C) , (C, F) , (G, D) , (D, H) . This clearly shows that the general connectivity is a global and accurate representation of connection among cells. In particular, all the procedures such as partitioning, pad-assignment, clustering and initial placement which are dependent on the representation of connection among cells will substantially benefit from the usage of the general connectivity.

Although the procedure for obtaining the k^{th} order connectivity between any two vertices is straight forward, the requirement on computational resources may be excessive if both k and the number of vertices are large. Fortunately, in most practical applications, it is adequate to limit k to within 5 [2, 4, 13]. Another merit of using the general connectivity is that it becomes easy to incorporate a critical path by treating the length of the critical path as less than k . In addition, it will be useful for clustering involving ring which is regarded as one of the most difficult problems in traditional clustering algorithms [3] [Figure 6].

For the first order improvement in the representation of the connections between cells, the general connectivity with the order $k = 2$ is sufficient. A simple algorithm to find the second order general connectivity is described as follows.

for each cell i and cell j , do
 $G(i, j, 2) = 0$;
 for each cell i , do

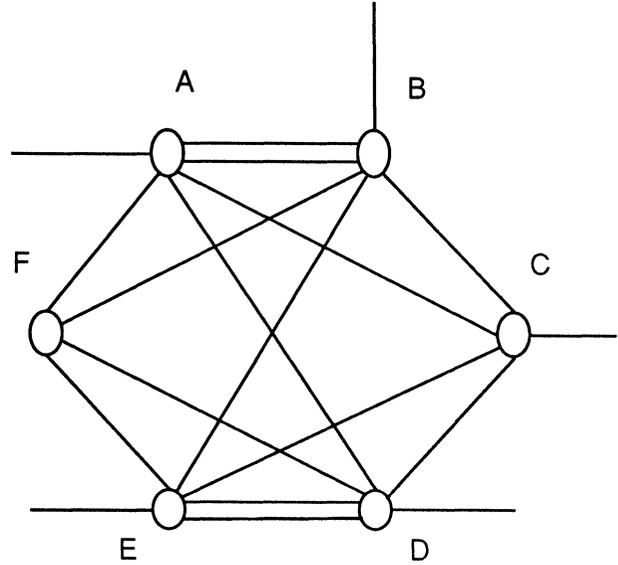


FIGURE 6 A clustering involving some rings.

for each cell j , where $i < j$ and $G(i, j, 1)$ is not equal to zero, do
 $G(i, j, 2) += G(i, j, 1)$;
 for each cell i , do
 for each cell n , do
 for each cell j , where $i < j$ and both $G(i, n, 1)$ and $G(j, n, 1)$ are not equal to zero, do
 $G(i, j, 2) += [G(i, n, 1) * G(j, n, 1)] / [G(i, n, 1) + G(j, n, 1)]$;

The complexity of this algorithm is $O(n^3)$, where n is the number of cells.

After the introduction of the concept of the general connectivity, we introduce another new concept of a stable cluster in Section 3.

3 STABLE AND NEAR STABLE CLUSTERING

Clustering is an important procedure carried out prior to the placement process [2, 4, 13]. In this paper, the general connectivity is used as a basis for the clustering process. We also introduce a new clustering concept, the 'stable cluster', to evaluate and improve the result of the clustering process.

Let X be a cluster, A a subset of X , $M = X - A$ be the complementary set of A in X , and T_A the connectivity between A and all the other cells (inside or outside of X),

$$T_A = \sum_{\forall r \notin A} G(A, r, 1).$$

Definition 6: The cluster X is a stable cluster if and only if for any A

$$G(A, M, 1) \geq T_A - G(A, M, 1)$$

$$A \neq \phi \text{ and } M \neq \phi.$$

Clearly, we call a cluster ‘stable’ if for any group of cells within the cluster, the connectivity between the group and the other cells within the same cluster is stronger than that between the group and all the cells outside the cluster. Hence, all the components of a stable cluster are so tightly connected to each other that the stable cluster can not be split. Clearly, a cluster can be inherently regarded as a cell in partition, placement or other layout phases only if the cluster is stable.

In traditional clustering approaches [3], the cell-pair with currently the largest connectivity is clustered and treated as a new basic cell in subsequent clustering. In this process, a constraint on cluster size is used to prevent the formation of excessively large clusters in early stages. The clustering process continues until all cells are included into a single cluster (the final cluster). The process of developing the final cluster is then used to obtain a linear placement strategy. Obviously, this approach does not guarantee that all the clusters obtained are stable.

The stable clusters are very useful for initial placement. However, to evaluate the stability of a given cluster, it is necessary to enumerate all possible subsets for the cluster. If there are m cells in the cluster, the computational complexity for the enumeration is $O(m!)$. Therefore, finding the stable clusters is an NP-hard problem. In order to develop an effective procedure to find the stable clusters within a reasonable computational time, it is necessary to introduce a more comprehensive heuristic clustering approach. We now present a theorem that facilitates the implementation of a heuristic algorithm for finding a stable cluster.

Theorem 1: A cluster X formed from two stable subclusters A and B , such that $X = A + B$, is a stable cluster if $G(A, B, 1) \geq T_A - G(A, B, 1)$ and $G(A, B, 1) \geq T_B - G(A, B, 1)$.

Proof of Theorem 1: Suppose C is any subset of A , and D is any subset of B . In order to prove that X is stable, it is necessary to show that

$$\begin{aligned} G[(C + D), (A - C + B - D), 1] &\geq T_{C+D} \\ &- G[(C + D), (A - C + B - D), 1]. \end{aligned}$$

Case 1. $C = A$ and $D = 0$.

In this case, $C + D = A$, $A - C + B - D = B$ and $T_{C+D} = T_A$.

Since $G(A, B, 1) \geq T_A - G(A, B, 1)$, hence, $G[(C + D), (A - C + B - D), 1] \geq T_{C+D} - G[(C + D), (A - C + B - D), 1]$.

(The proof for the case $C = 0$ and $D = B$ is similar to that for $C = A$ and $D = 0$).

Case 2. $C \neq 0$ and $D \neq 0$.

In this case,

$$\begin{aligned} G[(C + D), (A - C + B - D), 1] &= G[C, (A - C), 1] \\ &+ G[D, (A - C), 1] + G[C, (B - D), 1] \\ &+ G[D, (B - D), 1]. \end{aligned}$$

Since both A and B are stable,

$$\begin{aligned} G[C, (A - C), 1] + G[D, (A - C), 1] &+ G[C, (B - D), 1] + G[D, (B - D), 1] \\ &\geq T_C - G[C, (A - C), 1] + G[D, (A - C), 1] \\ &+ G[C, (B - D), 1] + T_D - G[D, (B - D), 1] \\ &= T_C - G[C, (A - C), 1] \\ &- G[D, (A - C), 1] - G[C, (B - D), 1] \\ &+ T_D - G[D, (B - D), 1] + R, \end{aligned}$$

where $R = 2\{G[D, (A - C), 1] + G[C, (B - D), 1]\}$.

Since $T_C + T_D = T_{C+D} + 2 \times G(C, D, 1) = T_{C+D} + S \geq T_{C+D}$, where $S = 2 \times G(C, D, 1)$,

hence, $G[(C + D), (A - C + B - D), 1] \geq T_{C+D} - G[(C + D), (A - C + B - D), 1] + R + S \geq T_{C+D} - G[(C + D), (A - C + B - D), 1]$.

[Q.E.D.]

From the proof, we can see that in order to ensure the stability of X , for case 1, the conditions $G(A, B, 1) \geq T_A - G(A, B, 1)$ and $G(A, B, 1) \geq T_B - G(A, B, 1)$ are necessary. However, in case 2, since the term $R + S$ is always non-negative, it is easy to see that the condition of both A and B being stable is sufficient but not necessary to ensure the stability of X . Intuitively, we now introduce a heuristic idea. If the conditions $G(A, B, 1) \geq T_A - G(A, B, 1)$ and $G(A, B, 1) \geq T_B - G(A, B, 1)$ are satisfied where the subclusters A and B are only ‘near stable’, the cluster X will still be stable in most situations. Based on this idea, we present a heuristic clustering algo-

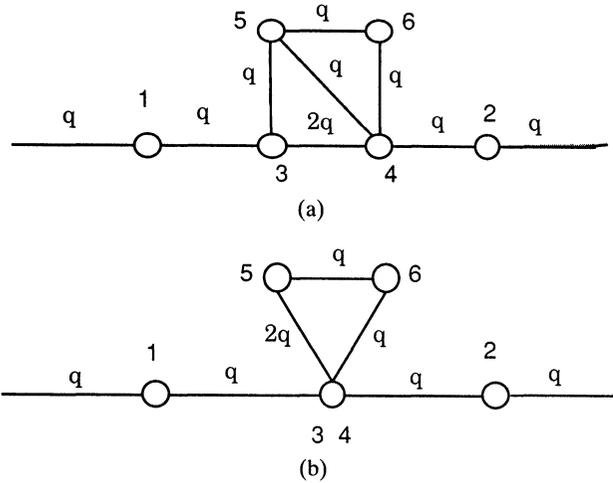


FIGURE 7 The reevaluation of the general connectivity.

rithm capable of producing stable clusters in most situations.

The algorithm consists of two parts. In the first part of the algorithm, a subcluster-pair A, B is clustered to form X if the conditions $G(A, B, 1) \geq T_A - G(A, B, 1)$ and $G(A, B, 1) \geq T_B - G(A, B, 1)$ are satisfied or the general connectivity value between A and B is not less than a prespecified value. The prespecified value is called 'immediate clustering threshold'. In other words, any subcluster-pair with lower general connectivity than the threshold will not be clustered. In the second part of the algorithm, an investigatory procedure back tracks the clustering process to check if the cluster developed in the first part of the algorithm is stable. The cluster X is considered to be a stable cluster if $G(A, B, 1) \geq T_A -$

$G(A, B, 1)$ and $G(A, B, 1) \geq T_B - G(A, B, 1)$ based on the heuristic idea; Otherwise, the cluster X is not stable and the subclusters A and B should be left in the decomposed state. The stability check continues until all the clusters are verified to be stable. A single cell is considered to be a stable cluster if it fails the stability checks with the other subclusters.

The cluster which includes only the original cells is called 'first level cluster'. The cluster is called $(n + 1)^{th}$ level cluster if the maximum level of its constituent subcluster is n . A clustering procedure is called *parallel clustering* if the $(n + 1)^{th}$ level cluster is formed only after all the n^{th} level clusters have been formed.

We now explain the need for the reevaluation of the general connectivity during the clustering process. Consider the circuit shown in Figure 7a where the weight of each of the connections is as indicated. In Figure 7a, $G(1, 2, 2) = 0$. After cell 3 and cell 4 are clustered, the result is shown in Figure 7b. Now, $G(1, 2, 2)$ is greater than zero. Hence, it is necessary to reevaluate the general connectivity of the cells related to the new cluster for subsequent clustering. Obviously, in parallel clustering, the reevaluation of the general connectivity is needed only at the starting of a new level.

After the clustering process has been completed, the maximum level of the clusters is called 'depth of the clustering tree'. The depth of the clustering tree should be as small as possible for a good placement [4]. Figure 8a shows a parallel clustering tree and Figure 8b shows a nonparallel clustering tree for the same set of cells. The depth of the clustering tree in Figure 8a is 3 and that in Figure 8b is 7. Obviously,

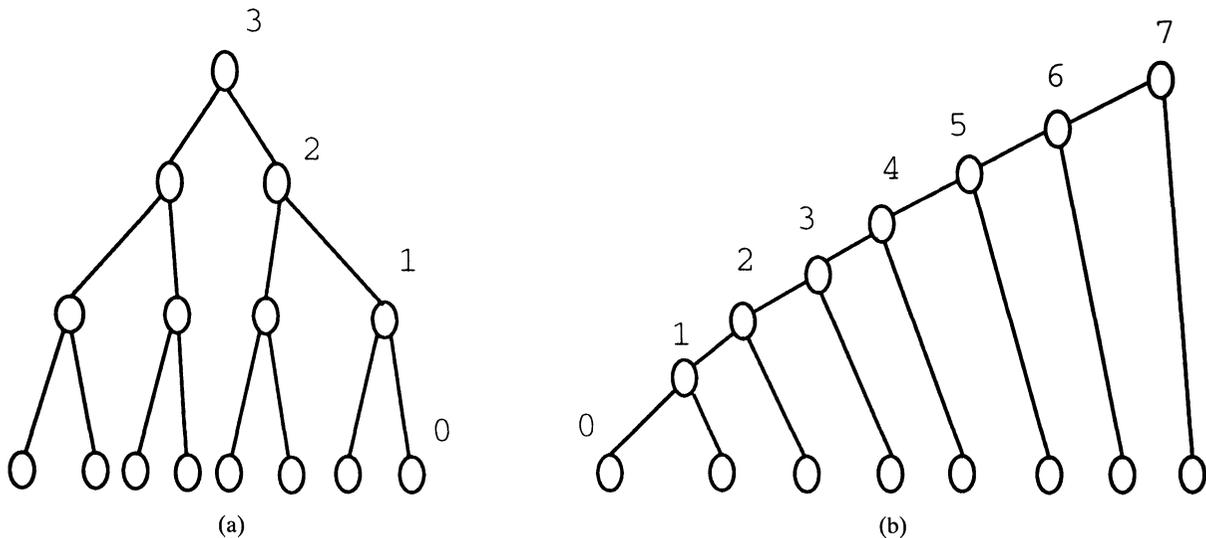


FIGURE 8 The clustering tree.

it is advantageous to use the parallel clustering to minimize the depth of the cluster tree. Therefore, the parallel clustering method is adopted in this paper. We shall also limit the number of cells in any cluster to be less than a maximum, N_{max} , to facilitate the placement process.

In the parallel clustering procedure, an important problem arises in that a cell which should become a member of an existing cluster X in the next clustering step is joined with another cell (or cluster) in the present step. The cell is then deprived of the possibility to cluster with X . In the worst situation, it would result in the two related clusters to be unstable. For instance, consider Figure 9, in which the connections and the associated weights are shown. Here we have chosen $N_{max} = 6$ and the immediate clustering threshold $= 4q$. In the example of Figure 9a, $G(A, B, 3) = G(I, J, 3) = 6.5q$, $G(D, G, 3) = 4q$, and $G(D, B, 3) = G(D, A, 3) = G(G, I, 3) = G(G, J, 3) = 4.9q$. It is obvious that the cluster formed by cells A, B, C, D, E and another cluster formed by cells F, G, H, I, J are stable. The key point is that cell D should be a member of the cluster including cell A and cell B , and cell G should be a member of the cluster including cell I and cell J . However, in the clustering procedure for the first level cluster, the cell-pair $D-G$ would be clustered as well as the cell-pair $A-B$ and cell-pair $I-J$ as $G(D,$

$G, 3) = 4q$ (Figure 9b). As a possible result, the cluster formed by cells A, B, C, D, G, E and another cluster formed by cells I, J would be constructed. But these two clusters would both be unstable. Therefore, in the parallel clustering procedure, another threshold called ‘potential clustering threshold’ should also be included to avoid the above problem. When two cells (or clusters) have been selected to cluster, we ‘lock’ all cells (or clusters) whose general connectivity with one of the selected cells is not less than the potential clustering threshold. In other words, the locked cells would be prevented from clustering with the other cells (or clusters) until the next clustering level. Hence, the locked cells (or clusters) can potentially join the previous cluster in the next level.

Based on the discussions in Section 2 and 3, we present a parallel clustering algorithm in Section 4.

4 ALGORITHM FOR THE PARALLEL CLUSTERING

A heuristic algorithm for parallel clustering based on the concepts of general connectivity and stable cluster can now be constructed. Before the commencement of the clustering process, the immediate clus-

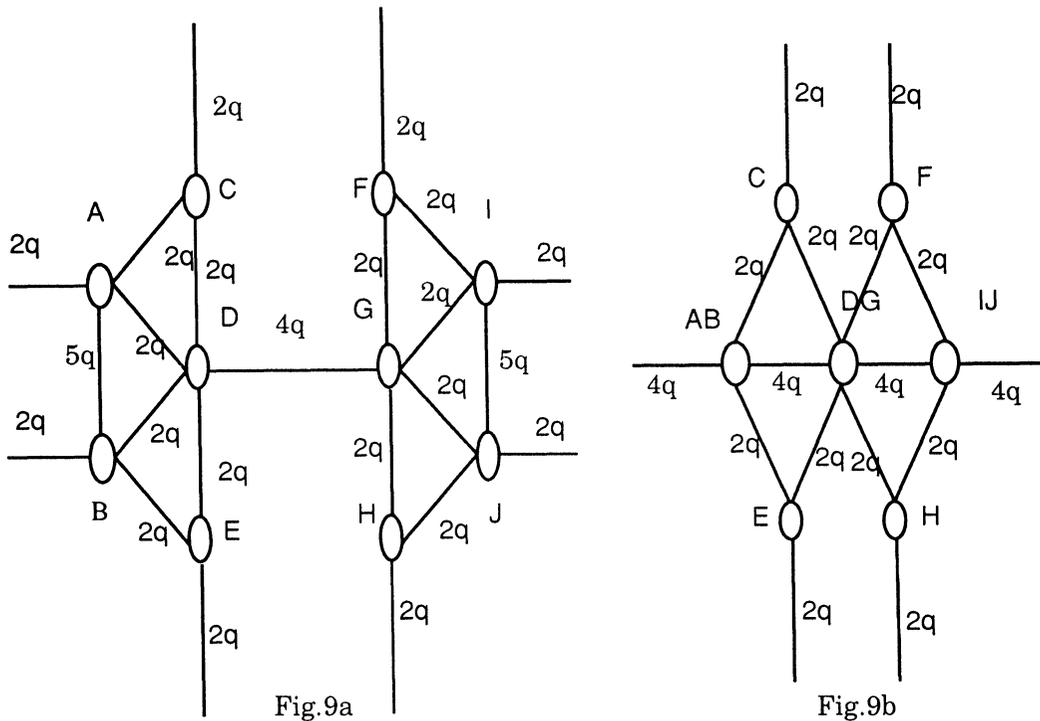


FIGURE 9 The example for the potential clustering threshold.

tering threshold (V_1), the potential clustering threshold (V_2) and the order of the general connectivity (k) are set. In the algorithm, 'cell' represents both the original cell as well as the clustered composite cell. The algorithm is as follows.

Step 1: Evaluate the general connectivity for all the cell-pairs (i, j) for the first level clustering. Otherwise, evaluate $G(i, j, k)$ just for the related cell-pairs. Clear all markings on the cells.

Step 2: Identify the unmarked cell-pair with the largest value of $G(i, j, k)$. Suppose the cell-pair identified is (r, s). If $G(r, s, 1) \geq T_r - G(r, s, 1)$ and $G(r, s, 1) \geq T_s - G(r, s, 1)$, or $G(r, s, k) \geq V_1$, go to step 3; else, go to step 5.

Step 3: Cell-pair (r, s) is marked '*immediately clusterable*', if the number of cells in cell-pair (r, s) is less than N_{\max} .

Step 4: For an unmarked cell t , if $G(r, t, k)$ or $G(s, t, k)$ is greater than V_2 , then a label '*potentially clusterable*' is marked on cell t . This locks cell t . Go to step 2.

Step 5: If there is no cell-pair marked with '*immediately clusterable*', go to step 7; else, go to step 6.

Step 6: All cell-pairs marked '*immediately clusterable*' are clustered to form new cells. The original cells are called the constituent cells and the resulting new cell is called a composite cell. All the relationships between the constituent and composite cells are recorded. Go to step 1.

Step 7: Clear the markings on all the composite cells.

Step 8: For each unmarked composite cell, the connectivity between the constituent cells A and B is reinspected for stability requirement. If $G(A, B, 1) \geq T_A - G(A, B, 1)$ and $G(A, B, 1) \geq T_B - G(A, B, 1)$, the composite cell is marked as a '*stable cell*'; otherwise, the composite cell is not stable and is decomposed into its constituent cells. Each of the constituent cell is now regarded as a composite cell if it is not an original cell.

Step 9: If there are no more unmarked composite cells, go to step 10; else, go to step 8.

Step 10: All the marked composite cells are considered as final clusters. Stop.

The '*potentially clusterable*' marking of step 4 essentially excludes cell t from forming a cluster with the leftover cells. Cell t is then free to cluster with the cluster formed by cell-pair (s, r) subsequently. In general, most of the clusters produced by the above algorithm are stable. Note that in the above procedure all the clusters are developed in parallel to minimize the depth of clustering tree. The complexity of this algorithm is $O(n^4)$ where $k = 2$ and

n is the number of cells. We shall present the results of the application of this algorithm in Section 5.

5 EXPERIMENTAL RESULTS AND DISCUSSION

The placement subsystem of the LSIS-II layout system presented in [5, 6] is used as a platform to evaluate the efficiency of the general connectivity concept and the parallel clustering approach. The LSIS-II layout system is a practical design system based on standard cells and macro cells, and has been installed in 10 institutions, universities and design houses. The original initial placement algorithm of the LSIS-II layout system used inside-outside connectivity as cell selection criterion. Cells are placed one at a time beginning from the periphery of a chip. Various iterations such as cell-movement, pairwise-interchange, neighbor-interchange, and cell-insertion are adopted to improve placement. Because of the application of multitarget objective function, the convergence curve for the improvement of the placement does not decrease monotonically (Figure 10). The algorithm is able to escape from being trapped at a local optimum solution to reach a better solution.

The initial placement subsystem is modified appropriately to accommodate the method presented in this paper. To examine the efficiency of the general connectivity and parallel clustering, no modification is made to the iterative placement procedure.

The circuits used in the experiments are ASICs taken from industries. The largest circuit HK5851 consists of about 28 thousands transistors. The results are tabulated in Tables I and II for comparison. It can be seen from Table I that the number of feed-throughs in the initial placements of circuits are reduced by 0.36%, 7.5%, and 19.4%, respectively. It can also be seen from Table II that the number of feed-throughs in the final placements are reduced by 5.2%, 9.4%, and 21.6%, respectively. In the above examples, the chip-lengths are reduced by 0.45%, 1.08%, and 3.49%, respectively. For all the placements, the number of blocks in each of the tested circuit is fixed.

Initial test runs show that this approach is very effective even when k is only 2, especially for large circuit. The improvement is even more significant for the larger and more complex examples. The number of feed-throughs in the largest circuit HK5851 is reduced by over twenty per cent and the length of one side of the chip is reduced by twenty grid spacing. This improvement is significant. Because of the re-

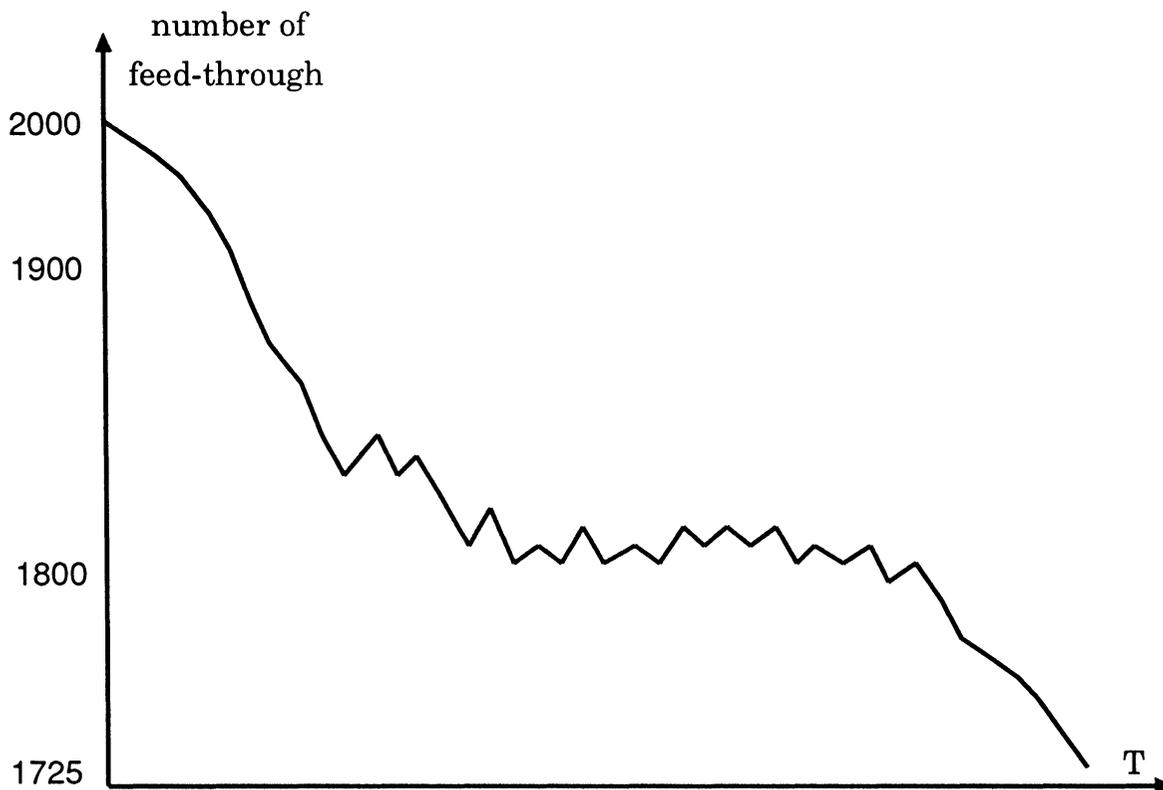


FIGURE 10 The convergency curve of placement improvement of HK5851.

TABLE I
The Comparison of Initial Placement Results

Project	Size of circuit		Number of feedthrough		CPU time (s)	
	No. cells	No. nets	original method	new method	original method	new method
HK5601	256	355	278	277	1.34	3.41
HK5852	608	699	946	875	3.75	21.3
HK5851	1724	1893	2818	2270	29.5	120.3

TABLE II
The Comparison of Final Placement Results

Project	chip-length			Number of feedthrough		
	original method	new method	improvement	original method	new method	improvement
HK5601	219	218	0.48%	269	255	5.20%
HK5852	372	368	1.08%	777	704	9.40%
HK5851	573	553	3.49%	2184	1712	21.6%

duced number of feed-throughs, the number of pins in each channel will also be reduced resulting in reduced channel routing densities. This will also result in reduced chip-height. The improvement of the final placement results is a strong indication to the importance of good initial placement in VLSI layout design. Another advantage of this method is that it

can be easily incorporated into any existing placement system.

The CPU time required by the initial placement (in the new approach, it includes the time required by the parallel clustering) are also listed in Table I. As can be seen from Table I, the computational time required by our new approach is slightly longer. This

is due to the time required by the clustering process. Nevertheless, the increase in the computational time is insignificant.

6 CONCLUSION

In this paper, we have presented a new concept of the general connectivity which yields a more precise and complete representation of the cell relationship. The equivalence between the connectivity graph and electrical network leads to a quantitative estimate of the general connectivity. We have also presented a heuristic parallel clustering approach based on the concepts of the general connectivity and stable cluster. We have shown its application in cell placement. The results of initial test runs are very encouraging. The approach leads to much better initial and final placement without much penalty in execution time. The potential of the general connectivity, the stable cluster and the parallel clustering approach in the layout improvement is excellent.

References

- [1] S. Goto, "A Two-Dimensional Placement Algorithm for the Master-Slice LSI Layout Problem," Proc. of 16th ACM/IEEE D.A. Conf., 1979, pp. 11-17.
- [2] M.Y. Yu and W.J. Zhuang, "The Optimal Construction of Clusters," Proc. of International Conference on Computer Aided Technology, 1988, Hong Kong.
- [3] Dotoald M. Schuler et al. "Clustering and Linear Placement," Proc. of 9th D.A. Workshop, 1972, pp. 50-56.
- [4] M.Y. Yu, X.L. Hong, Y.E. Lien, Z.Z. Ma, J.G. Bo, and W.J. Zhuang, "A new Clustering Approach and Its Application To BBL Placement," Proc. of European D.A. Conf., 1990, pp. 665-669.
- [5] W.J. Zhuang, K.X. Cheng et al, "LSIS-II Automated Layout System," Chinese Journal of Semiconductors, Vol. 8, No. 5, 1987, pp. 270-276.
- [6] K.X. Cheng and W.J. Zhuang, "The Placement Subsystem of LSIS-II Automatic Layout System," Chinese Journal of Semiconductors, Vol. 7, No. 4, 1986, pp. 412-417.
- [7] S. Kirkpatrick et al, "Optimization by Simulated Annealing," Science, Vol. 220, 4598, 1983, pp. 671.
- [8] S. Nahar et al, "Simulated Annealing and Combinatorial Optimization," Proc. of 23th ACM/IEEE D.A. Conf., 1986, pp. 293.
- [9] C. Sechen et al, "An Improved Simulated Annealing Algorithm for Row-Based Placement," Proc. IEEE of ICCAD, 1987, pp. 478.
- [10] S. Mallela et al, "Clustering based Simulated Annealing for Standard Cell Placement," Proc. of 25th ACM/IEEE D.A. Conf., 1988, pp. 312-317.
- [11] P.S. Tang, "The Theory and Method of VLSI CAD," Fu Dan University Publishing House, 1990, pp. 257-259.
- [12] Somchai Prasitjutrakul, "Performance-Driven Chip Floorplanning and Global Routing," Report No. UIUCDCS-R-91. 1689, University of Illinois at Urbana Champaign, 1991.
- [13] W.M. Dai and E.S. Kuh, "Simultaneous Floor Placement and Global Routing for Hierarchical Building Block Layout," IEEE Trans. on CAD, Vol. CAD-6, No. 5, 1987, pp. 828-837.
- [14] W.E. Donath, "Complexity Theory and Design Automation," Proc. of the 17th Design Automation Conference, 1980, pp. 412-419.
- [15] B.T. Preas et al, "Automatic Placement: A Review of Current Techniques," Proc. of 23th Design Automation Conference, 1986, pp. 622-629.
- [16] S. Goto et al, "Suboptimal Solution of the Backboard Ordering With Channel Capacity Constraint," IEEE Trans. on CAS, CAS-24, 1977, pp. 645.

Biographies

WENJUN ZHUANG was born in Shanghai China on May 10, 1946. He graduated from Tsinghua University, Beijing, China, in 1968. From 1970 to 1989, he worked in the Institute of Semiconductors, Chinese Academy of Sciences, where he became a professor in 1988. From 1989 to 1992, he had been a professor and director of ICCAD Department of Beijing Sofaware Laboratory, Chinese Academy of Sciences. Since March 1992 he has been a senior fellow at National University of Singapore. Since 1970 he has engaged in IC device test, computer design, ICCAD and software fault tolerance. Wenjun Zhuang served as a member of National ICCAD Specialist Committee of China.

NENG YAN received the B.S. degree in Computer Science from Tsinghau University in 1987, and the M.S. degree in ICCAD from Chinese Academy of Sciences in 1990. Since then, he joined Beijing Software Laboratory, Chinese Academy of Sciences. His research interests include design automation of VLSI, especially, Clustering, placement and pin/pad assignment.

YONGCHING LIM was born in Malaysia. He received the ACGI and B.Sc. degrees in 1977 and the DIC and Ph.D degrees in 1980, all in Electrical Engineering from Imperial College, University of London. From 1980 to 1982, he was a National Research Council Research Associate in the Naval Postgraduate School, Monterey, U.S.A. He joined the Department of Electrical Engineering, National University of Singapore in 1982. His research interests include VLSI circuits and System design, digital signal processing, and design automation. He served as an Associate Editor for the IEEE Transactions on Circuits and Systems from 1991 to 1993 and is an Associate Editor for Circuits, Systems and Signal Processing since 1993.

GANESH SAMUDRA was born in India. He received the M.Sc. from Indian Institute of Technology, Bombay in 1979 and M.S., M.S.E.E., and Ph.D degrees from Purdue University. In 1985, he joined Texas Instruments, Inc., Dallas where he worked on development and support of semiconductor process and device simulation. He was elected Member of Group Technical Staff in 1988 for outstanding technical contributions in simulation. Since 1989, he is with the Department of Electrical Engineering, National University of Singapore where he is presently a senior lecturer. He is presently working in Technology and Design CAD tools. Dr. Samudra is chairman of IEEE Circuits and System Chapter of Singapore.

