HYBRID CIRCUITS PLACEMENT

Jing Lee

Department of Electronic Engineering

Southern Taiwan University of Technology

Tainan, Taiwan 701, R.O.C.

Email: leejing@mail.stut.edu.tw

Hybrid circuits are fabricated on an insulating ceramic substrate (alumina and beryllia) utilizing some combination of thick- or thin-film components, monolithic semiconductor devices, and discrete parts. Sealing of the hybrid is usually performed in a dry nitrogen atmosphere. The packages are hermetically sealed by resistance welding of the lid to the case [Jon82]. Advances in the design of complex hybrid circuits have yielded devices with many components, some of which may dissipate appreciable power. This can result in higher operating temperatures if the dissipated heat is not properly removed. It is well known that most of the physical and chemical processes that can cause component failure are usually accelerated at elevated temperatures [Art84, Lyc80]. In addition, the temperature variation on a substrate may affect system reliability as it can induce thermal stresses [Wal83]. When the stresses are severe enough and go through enough cycles, they can cause the semiconductor device to fail, usually by rupturing of the solder joints. Since the temperature bound plays a critical role in the safe operating region of all electronic components, it is desirable to maintain a low device operating temperature, and to minimize component to component temperature variations.

Thermal designs are strongly influenced by the component placement. The temperature profiles of the substrate as well as the junction temperature of each component depend heavily on its placement. Furthermore, the placement also affects the circuit wireability. A pool placement not only occupies larger areas and degrades performance but also leads to a difficult or even impossible routing task. Therefore, in order to obtain a better placement, one must take reliability and wireability into consideration at the placement stage. However, due to the complexity of the problems, the placement for reliability and wireability has traditionally been done separately. The primary goal of placement has been focused on improving routability [Oda87, Qui79, Sia87, Sch72, Fid82, Che92]. Thus, the criteria for success have been based on either minimizing the total wiring lengths among the interconnected components [Oda87, Qui79, Sia87], or reducing the maximum congestion on the board [Sch72, Fid82, Che92]. Recently, some studies have been focused on the improving the system reliability for power hybrid circuits (PHCs) [Wip82] and for convectively cooled printed circuit boards (PCBs) [Pec87, Dan89, May78, Ost92, Eli90]. In addition, Osterman and Pecht [Ost90] developed a placement procedure for both reliability and wireability of convectively cooled PCBs.

In this study, a **hierarchical** placement procedure is presented to take care of both the reliability and wireability for PHCs, where the components are assumed to have a fixed size. The components are grouped into **hot** and **cool cells** according to their dissipated power. Hot cells generate obvious heat; cool cells generate negligible heat. In addition, each I/O pad is given the same status as a cell, namely **I/O cell**. Then each possible pad location is defined to be a block of a type which permits only I/O type cells. The placement strategy for hot cells is to uniformly distribute the heat with min-cut objective, and that for cool cells is to minimize the total wiring lengths of signal nets.

The current placement algorithms can be grouped into four classes : sequential optimization methods, global optimization methods, partitioning methods, and iterative improvement methods. Each approach is discussed briefly in the following.

The sequential optimization techniques assign the cells onto a carrier step by step based on an incomplete placement. This type of algorithms has been widely used for initial placement for wireability [20, 21]. However, these algorithms are improper for our problem. It is because that the heat-conduction equation is an elliptic type of partial differential equation, the thermal energy generated by any cell will transfer to all the other cells. The temperature distribution of a carrier therefore can not be determined unless the cell placement had been determined. So, it is impossible to choose the best placement for reliability in several alternates.

The global method presented by Pecht and Naft [18] has succeed to the placement with reliability and wireability for convectively cooled PCBs. In their approach, they first constructed a position-adjacency matrix which was mapped from the optimal placement in reliability. Then, they combined the position-adjacency matrix with the connectivity matrix, constructed from the signal nets by a weighting factor, to form a hybrid matrix. After that, the force-directed placement technique was used to obtain the relative positions of the modules based on the hybrid matrix. However, since the generated heat is removed mainly by conduction in the package level for hybrid circuits, no presented method can obtain the optimal placement for hybrid circuit even only to optimize reliability. So, Pecht and Naft's approach also can not be used for our problem.

Partitioning algorithms [9-11] consider all interconnections in parallel

and then move the components in steps by partitioning the given circuit and layout area into two or more partitions. The process is carried out till each subcircuit consists of a single component and has a unique place on the layout area. During partitioning, the number of nets that are cut by the partition is usually minimized. These methods are restricted such that all vertices must have even sizes or dissipated heat. However, it is not true for our problem since the components can have different sizes and dissipated powers in hybrid circuits. So, these methods also are improper here.

Iterative improvement algorithms iteratively transform a complete solution into an improved, complete solution. These methods had been widely used for many combinatorial problems [22-27]. For multi-objective optimization problems, these methods have to combine all objective functions into a hybrid objective function. However, it is difficult to combined the reliability with the wireability into a hybrid objective function. So, these methods are difficult to be used for our problem. In addition, even consider system reliability as the only objective function for placements, the iterative improvement algorithms are still too time-consuming, since no existed method can choose a better one in two different placements without calculating their temperature distributions.

Traditionally, the hybrid circuit placement is mainly performed manually. We have looked into placement processes for this problem by skillful designers. They take care of the structure and the function of the circuits from the global point of view. They partition the whole circuit into several groups, considering the function, the structure, and the heat dissipation of the circuits. The partitioned blocks are placed on a carrier taking account of the heat dissipation and the wirings of signals among them. The cells generated large heats are divided apart on a carrier for avoiding locally high heat flux region.

There are three phases in the present placement method: placing the relative positions of the hot and I/O cells in phase 1, placing the relative positions of the cool cells in phase 2, and determining the real layout in phase 3.

5.1 Placing Hot And I/O Cells

The kernel scheme in this phase is the **clustering growth algorithm** which had been introduced in CHAPTER 3. The algorithm can partition a hypergraph (i.e., an abstract circuit model) into several sub-hypergraphs with min-cut objective.

There are two stages in this phase. They are hot cell growth stage and hot cell assignment stage.

5.1.1 Hot Cell Growth Stage

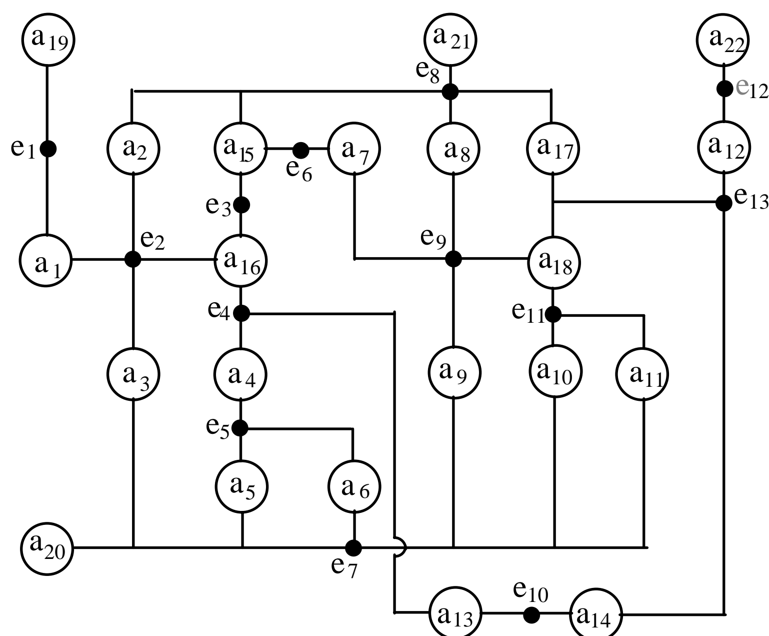
Here, the cells and nets are modeled by vertices and hyperedges, respectively. The vertex that models a hot or cool cell has a size weight of one; the vertex that models an I/O cell has a size weight of zero. So, the size of a cluster is equal to the number of hot and cool cells inside the cluster. The **capacity** of a cluster is proportional to its dissipated power. A shunk hypergraph is obtained by executing

$$H \leftarrow \text{call Clustering}(H, n)$$

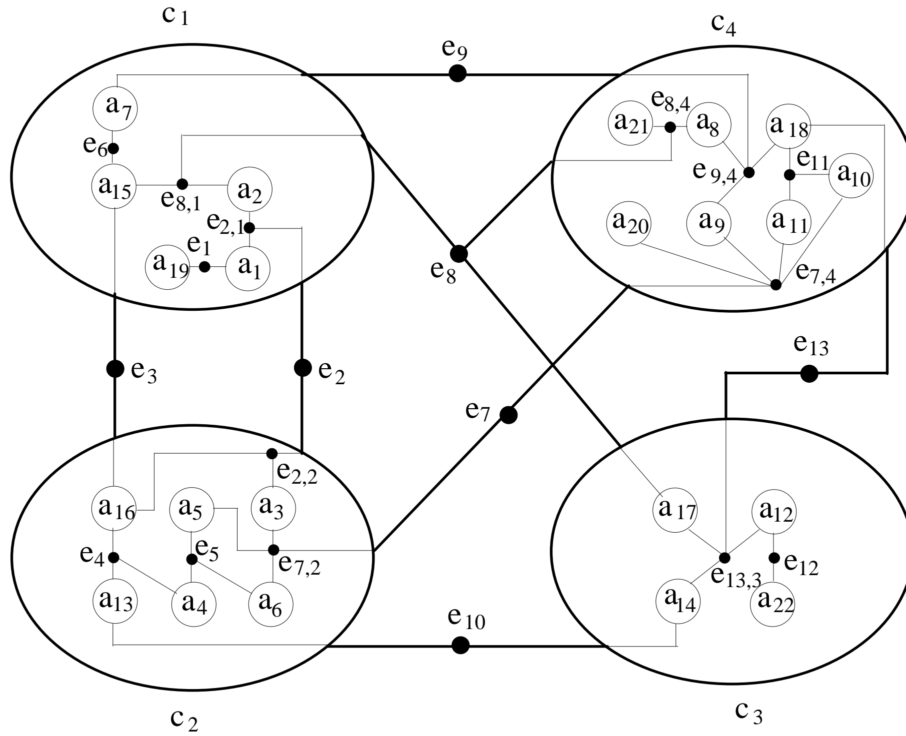
where n is the number of hot cells. The **seeds** of clusters are the hot cells. In this shunk hypergraph, each vertex is a cluster of one hot, and several I/O and cool cells. An example is shown in Figs. 5.1(a) and (b). As depicted in Fig. 5.1(a), the example circuit has 14 cool cells (indexed by a_1, a_2, ..., a_{14}), 4 hot cells (indexed by a_{15}, a_{16}, a_{17}, a_{18}), 4 I/O cells (indexed by a_{19}, a_{20}, a_{21}, a_{22}), and 13 nets. The dissipated power ratio is a_{15}:a_{16}:a_{17}:a_{18} = 3 : 5 : 2 : 6. After the hot cell growth stage, one can obtain a shunk hypergraph as shown in Fig. 5.1(b).

5.1.2 Hot Cell Assignment Stage

In this stage, a carrier is defined by a vector, S= (ℓ, w, x, y), where ℓ and w are the length and width of the carrier, respectively; x and y are the coordinates of the center of the carrier. The shunk hypergraph obtained by the hot cell growth stage and its corresponding carrier are recursively quadrisected until each vertex is partitioned with other vertices by the following procedure:



(a) A hypergraph with 22 vertices and 13 nets.



(b) The shunk hypergraph and four subhypergraphs after executing the clustering algorithm.

Fig. 5.1 An example to describe the hot cell growth stage.

procedure HOT_CELL_ASSIGNMENT(H, S)
// Input a shunk hypergraph with a carrier. Output the positions of hot cells. //
if $|V| = 1$ **then** [Output(V, x, y); **exit**]
else $(H_1, S_1, H_2, S_2) \leftarrow$ **call** BISECT(H, S)
endif
$(i, j) \leftarrow (3,6)$
if $|V_1| = 1$ **then** [$i \leftarrow 5$; Output(V_1, x_1, y_1)]
else $(H_3, S_3, H_4, S_4) \leftarrow$ **call** BISECT(H_1, S_1)
endif
if $|V_2| = 1$ **then** [$j \leftarrow 4$; Output(V_2, x_2, y_2)]
else $(H_5, S_5, H_6, S_6) \leftarrow$ **call** BISECT(H_2, S_2)
endif
while $i \leq j$ **do** [$i \leftarrow i+1$; **call** HOT_CELL_ASSIGNMENT(H_i, S_i)] **repeat**
if there are some I/O cells that uncertain their positions **then** assign them to I/O pads subject to minimize the total wiring lengths **endif**
end HOT_CELL_ASSIGNMENT

where BISECT( ) is the bisection procedure described as follows:

procedure BISECT(H, S) // Bisection a hypergraph with its carrier //
$(H_1, H_2) \leftarrow$ **call** CLUSTERING(H, 2)
if $w \leq \ell$ **then** the cut line is vertical
else the cut line is horizontal
endif
Compare two different assignments and **select** the one of less half-perimeter length.
Position the clusters at the centers of their corresponding half-carriers.
Assign the positions of I/O cells.
Output(H_1, S_1, H_2, S_2).
end BISECT

The quadrisection stage is shown in Fig. 5.2. For each quadrisection pass there are at most three bisection procedures. For each bisection course, the hypergraph is partitioned into two sub-hypergraphs of evenly heat dissipation by our clustering algorithm, and its corresponding carrier is also divided into two sub-carriers of the shape as square as possible by a cut line. The area ratio of the sub-carrier is proportional to the dissipated power of its corresponding sub-hypergraph. Then, the two sub-hypergraphs are assigned into the corresponding carrier subject to minimization of the total wiring lengths of the shunk hypergraph. Here, each cluster is positioned at the center of its corresponding carrier and the total wiring lengths are estimated by the summation of the total half-perimeter lengths.

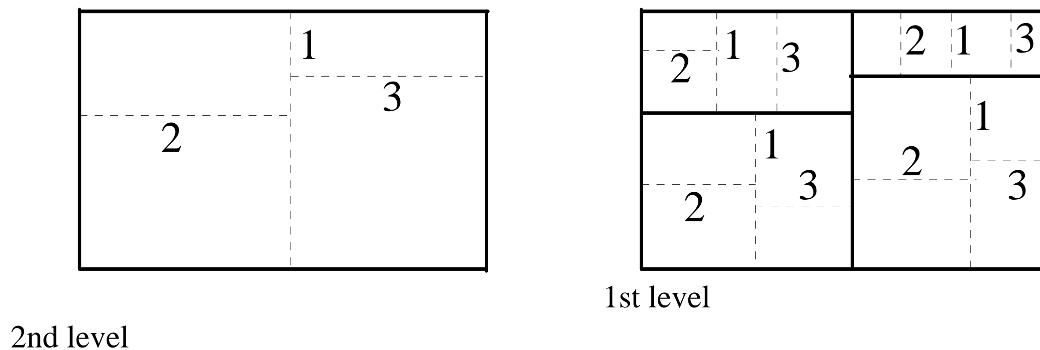


Fig. 5.2 Quadrisection stage

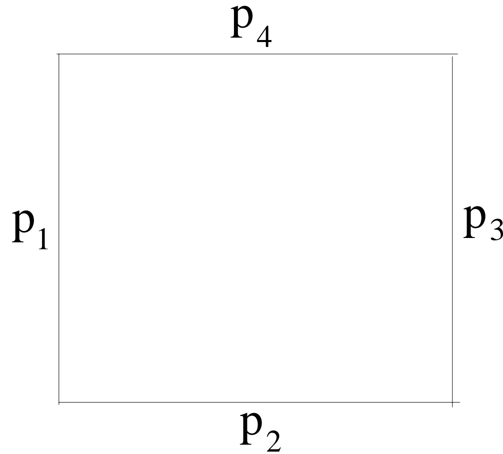


Fig. 5.3 A square carrier with four I/O pads. S = (80, 80, 40, 40)

Next, the I/O cells are assigned to I/O pads. Consider the shunk hypergraph shown in Fig. 5.1(b) to be placed on a square carrier with four I/O pads as shown in Fig. 5.3. The first bisection is shown in Fig. 5.4, where the hypergraph was partitioned into two sub-hypergraphs, H_1 and H_2, with the same dissipated powers. A vertical cut line divides the carrier into two parts of equal area; H_1 into the left and H_2 into the right. Since V_1 includes only one I/O cell, a_{19}, a_{19} apparently must be assigned into p_1. In this stage, the positions of the other three I/O cells are undetermined, they need to be further decided. In the second bisection, a horizontal cut line partitioned the left half carrier into two parts with the area ratio of 3:5. There are two possible forms as shown in Fig. 5.5(a) and (b) for this partitioning. Since they have the same total wiring lengths, one can select each one for this bisection. In the third bisection, Figs. 5.6(a) and (b) show two different forms for dividing the right half carrier. One can see that Fig. 5.6(b) has less total wiring lengths than Fig. 5.6(a), so the cut line as shown in Fig. 5.6(b) is better than in Fig. 5.6(a). After finish this bisection the position of a_{22} that belongs to c_3 can be assigned to p_2. After finishing the quadrisection procedure, each hot cell is assigned to at the center of its corresponding sub-carrier. If some I/O cells still uncertain their positions, they will be assigned to I/O pads subjected to minimization of their total wiring lengths. So, a_{20} and a_{21} are assigned into p_4 and p_3, respectively.

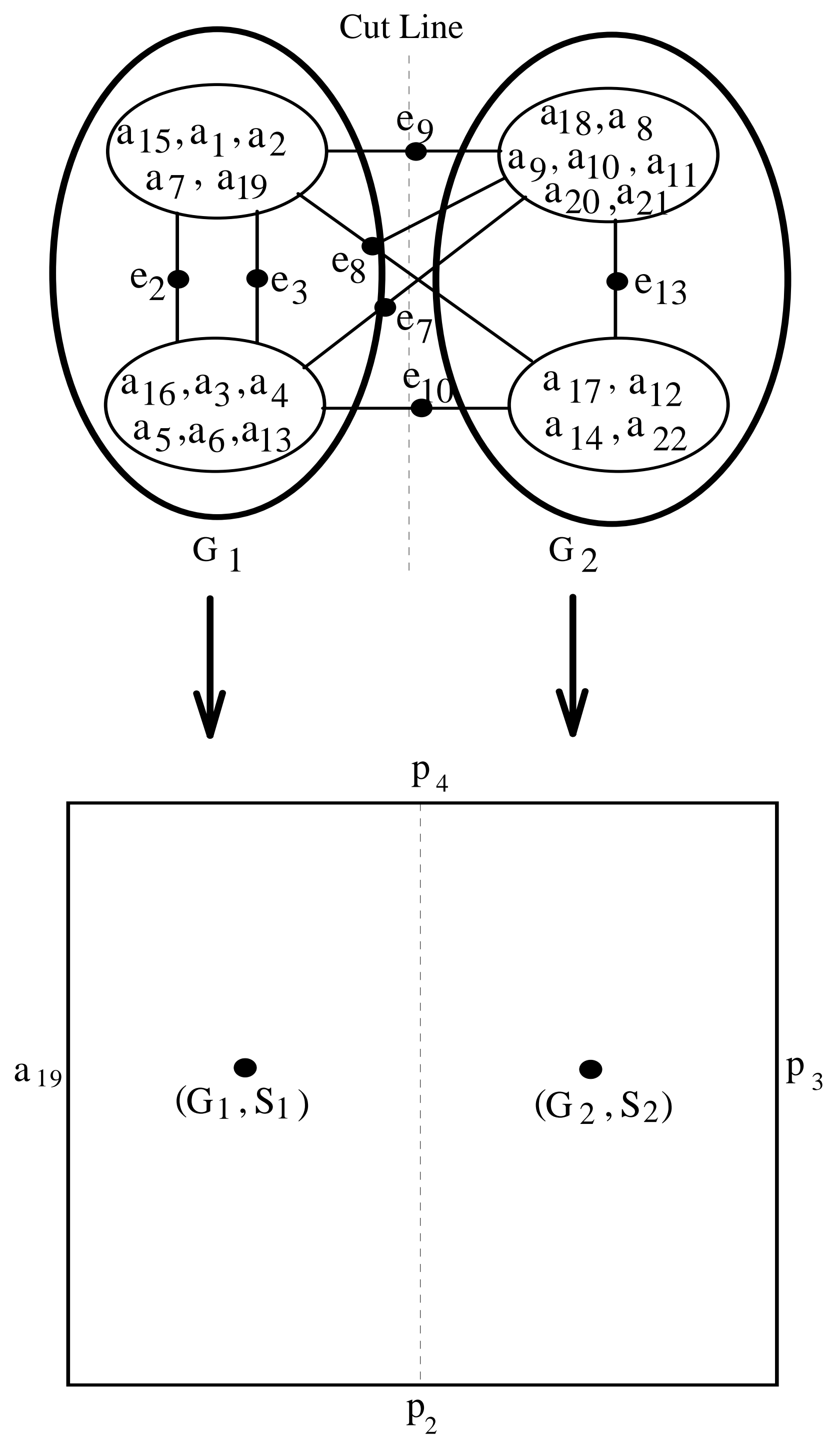
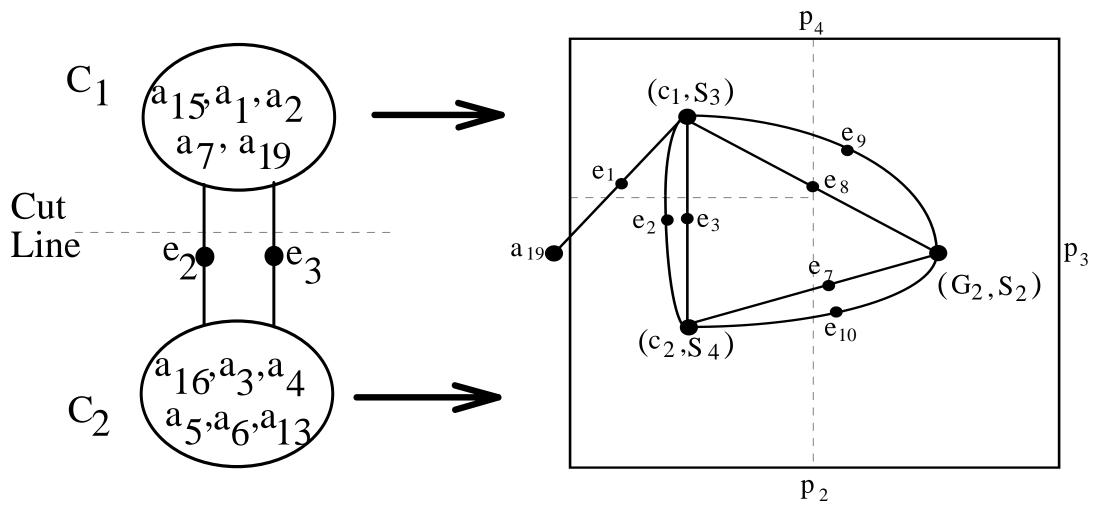
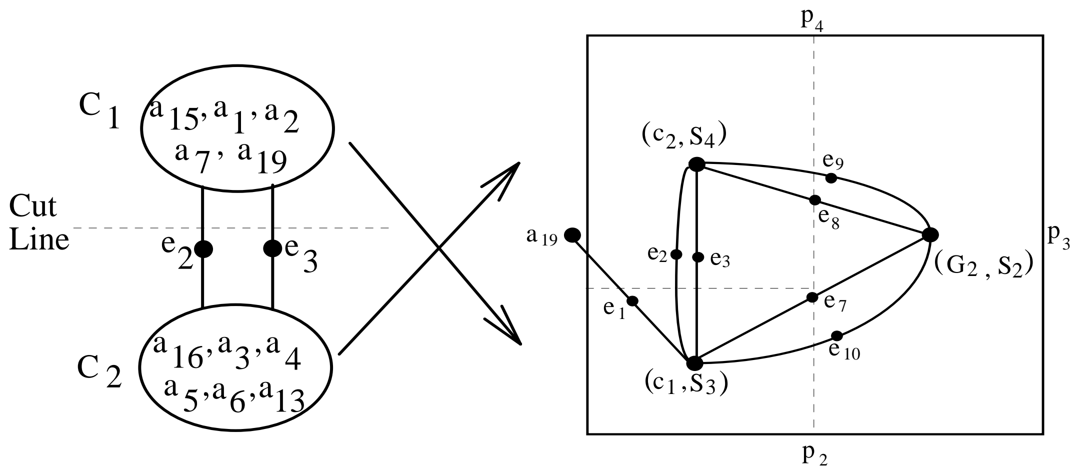


Fig. 5.4 The first bisection. $V_1 = \{c_1, c_2\}$; $V_2 = \{c_3, c_4\}$. $S_1 = (80, 40, 20, 40)$ and $S_2 = (80, 40, 60, 40)$.



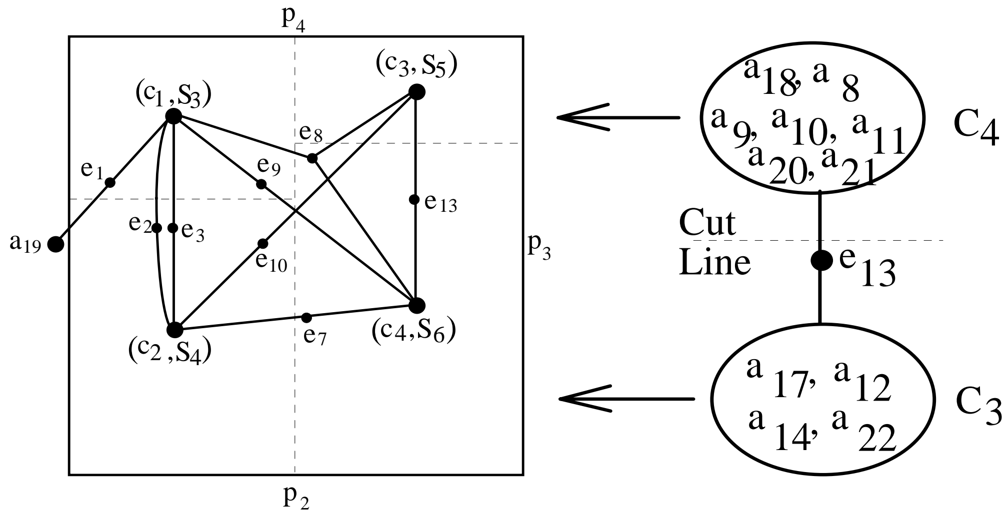
(a) Half-perimeter length = 365 unit. S_3=(40, 30, 20, 65); S_4=(40, 50, 20, 25).

(continue)

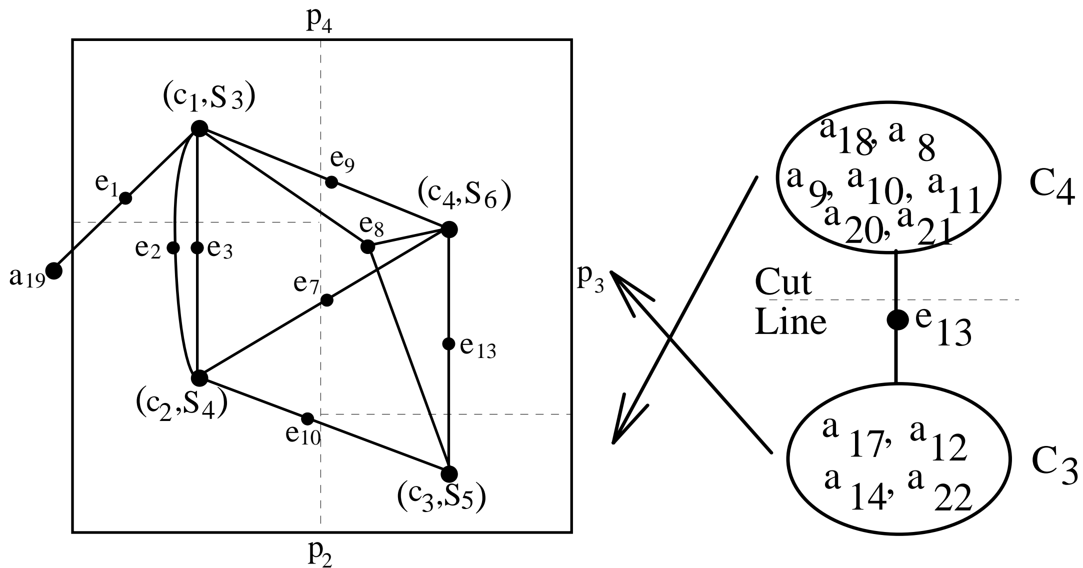


(b) Half-perimeter length = 365 unit. S_3=(40, 30, 20, 15); S_4=(40, 50, 20, 55).

Fig. 5.5 Two possible partitions for the second bisection.



(a) Half-perimeter length = 450 unit. S_5=(40, 20, 60, 70); S_6=(40, 60, 60, 30).



(b) Half-perimeter length = 435 unit. S_5=(40, 20, 60, 10); S_6=(40, 60, 60, 50).

Fig. 5.6 Two possible partitions for the third bisection.

The quality of the partitioning results obtained by our clustering algorithm depends on the seeds' selection. We found that good seeds must satisfy the following characteristics：

1) Seeds must be uniformly distributed throughout the hypergraph.
2) The vertex with the most dissipated power has a priority to be selected as a seed.

A seed generator with the above characteristics had been presented in CHAPTER 3.

We do not use top-down approaches, such as those presented in [Che92, Kri84], for the two-way partitioning of a hypergraph. These methods are restricted such that all vertices must have even sizes or dissipated heat. However, in a shunk hypergraph this constraint is unsatisfied since the vertices can have different sizes and dissipated powers. So, top-down methods are improper here.

5.2 Placing Cool Cells

The **force-directed placement algorithm (FDA)** presented in [Qui79] is used to position the cool cells in this phase. In this method, a force-model is used to determine the state of equilibrium. Hook's Law gives attraction forces for cells connected by nets, and repulsive forces for those which are not connected.

Here, circuits are modeled by a graph. Each cell is considered as a vertex and each net with n pins is simulated by a clique with edge weight of $4/(n^2 - \mathrm{mod}(n,2))$. If two vertices are connected by more than one edge, these edges are reduced to one edge with the sum of the individual edge weight. If two vertices v_i and v_j are linked by an edge with weight, K_{ij}, in the graph, there is an attractive force proportional to the weight times the distance between them. On the other hand, there are repulsive forces among unconnected vertices. The coefficient of the repulsive force between v_i and v_j, R_{ij}, is defined by

$$R_{ij} = \begin{cases} 0, & \text{for either } i = j \text{ or } K_{ij} \neq 0; \\ \dfrac{1}{U \times C_r} \sum_{i=1}^{k} \sum_{j=1}^{k} K_{ij}, & \text{otherwise.} \end{cases}$$

where U is the number of unconnected pair and C_r is a controlling parameter. Let x_i and y_i be the position of v_i. The force exerted on v_i by v_j therefore is

$$F_{ij,x} = -K_{ij} \times \Delta x_{ij} + R_{ij} \times \frac{\Delta x_{ij}}{\Delta s_{ij}} \qquad \text{in x direction,}$$

and

$$F_{ij,y} = -K_{ij} \times \Delta y_{ij} + R_{ij} \times \frac{\Delta y_{ij}}{\Delta s_{ij}} \qquad \text{in y direction,}$$

where $\Delta x_{ij} = x_j - x_i$, $\Delta y_{ij} = y_j - y_i$, and $\Delta s_{ij} = |\Delta x_{ij}| + |\Delta y_{ij}|$.

The hot and I/O vertices are fixed since their coordinates have been decided in phase 1. On the other hand, cool vertices are moveable and their locations need to be decided. If there are a total of N vertices and M moveable vertices, then for i from 1 to M,

$$F_{i,x} = \sum_{j=1}^{N} \left(-K_{ij} \times \Delta x_{ij} + R_{ij} \times \frac{\Delta x_{ij}}{\Delta s_{ij}} \right)$$

$$F_{i,y} = \sum_{j=1}^{N} \left(-K_{ij} \times \Delta y_{ij} + R_{ij} \times \frac{\Delta y_{ij}}{\Delta s_{ij}} \right)$$

The set of simultaneous equations is solved by setting $F_{i,x}$ and $F_{i,y}$ to zero and solving for x_i and y_i. A modified Newton-Raphson method [Qui79] is used to solve this system of equations to find the correct relative position of every cell with respect to every other cell. Fig. 5.7 shows the relative placement of the sample circuit after this phase.

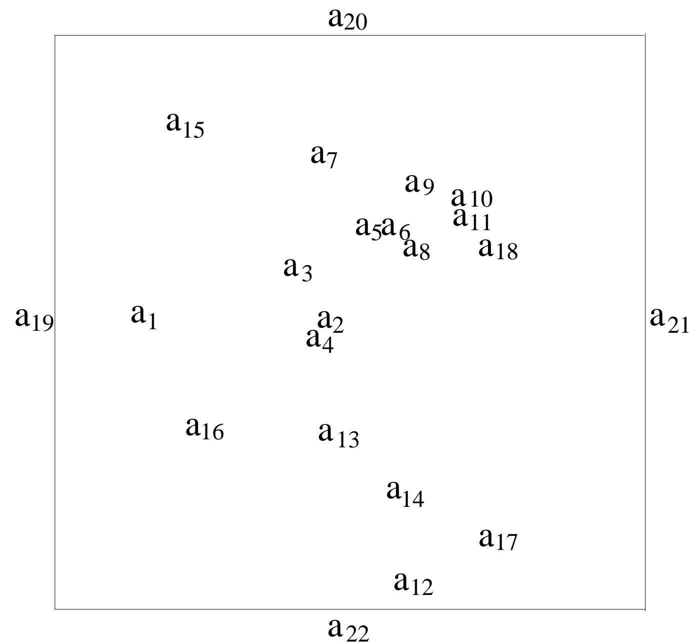


Fig. 5.7 Relative placement.

5.3 Real Layout

The relative placement is further translated to real layout. An interactive graphics package program. This system mainly includes these functions:

1) It displays the positions of cells on the screen. The user can move and fix any particular cell to any position on the substrate. While moving a cell, the program displays the variations of the placement on the screen. The new relative positions of the unfixed cells are obtained by FDA.

2) The reliability of the hybrid circuit is also displayed on the screen. The temperature distribution of the cells is obtained by the TAMS program [Ell83].

3) The program estimates the total routing lengths by the sum of the half-perimeter length for each net as a measure of the wireability of a placement.

In this manner the user can easily find good final positions for all cells.

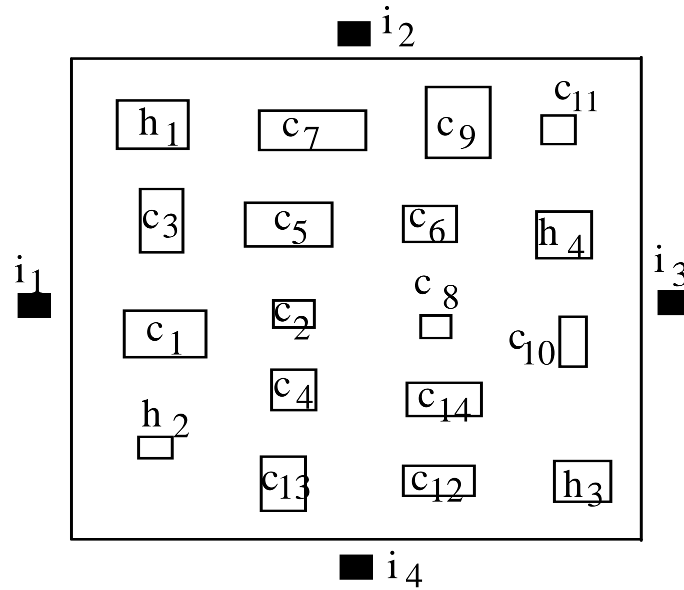


Fig. 5.8 Real placement from Fig. 5.7

5.4 **Computation Examples**

To examine the placement method discussed above two examples were tested and the results are compared to those obtained by FDA. The first example includes 9 hot cells, 33 cool cells, 6 I/O pads and 48 nets. The dissipated powers of the hot cells were one 1W-cell, five 1.5W-cells, one 2W-cell, and two 3W-cells. The second example includes 9 hot cells, 49 cool cells, 22 I/O pads, and 80 nets. The dissipated powers of the hot cells were one 0.6W-cell, one 1.4W-cell, two 1.5W-cells, one 2.5W-cell, two 3W-cells, and two 6W-cells. Each PHC package has an aluminum heat sink epoxied to a ceramic (Al_2O_3) base. The packages were cooled by air at a velocity of 2.54 m/sec.

The failure rate of a cell was calculated by using the Arrhenius formula

$$\lambda = \lambda_o \times \exp\left[\frac{E_a}{k} \times \left(\frac{1}{298} - \frac{1}{T}\right)\right]$$

where λ and λ_o are the failure rates at T K and at 298 K, respectively; E_a is the activation energy (eV); k is Boltzmann's constant. To determine the system failure rate, various operating parameters had to be specified. Without loss of generality, we assumed all cells to be the same factors of λ_o and E_a, which were 0.001 failures per million hours) and 1 eV, respectively. The system failure rate is calculated as the sum of individual failure rates.

Figs 5.9(a) and 5.10(a) show the placement results with temperature distribution of example 1 and 2 obtained by FDA, respectively. Figs 5.9(b) and 5.10(b) show the results obtained by our method. Key results are tabulated for comparison in Table 5.1. One can see that our results have lower T_{max} (i.e., the highest temperature in all components) and higher T_{min} (i.e., the lowest temperature in all components). Our substrates therefore have more uniform temperature distribution. The maximum temperature variation is reduced by 14 ^{o}C and 13 ^{o}C when compared to those by FDA for examples 1 and 2, respectively. The results also show that our method improves the reliability by 28.5 % and 38.5 % for examples 1 and 2, respectively. Since our algorithm is a tradeoff between reliability and routability, thus following the improvement in reliability, our algorithm results in 9% and 10.7 % increase in total wiring lengths than the results by FDA for examples 1 and 2, respectively.

Table 5.1 **The comparisons between FDA and our method**.

Items	Example 1		Example 2	
	FDA	Ours	FDA	Ours
T_{max}	126 ℃	118 ℃	135 ℃	126 ℃
T_{min}	80 ℃	86 ℃	60 ℃	64 ℃
ΔT_{max}	46 ℃	32 ℃	75 ℃	62 ℃
Wire Length	992	1081	1600	1771
λ_T	133.0 f / M	95.1 f / M	89.2 f / M	54.9 f / M

※ $\Delta T_{max} = T_{max} - T_{min}$.

5.5 Summary

While a PHC laid out for optimizing the reliability could be unroutable or extremely time consuming, a PHC laid out for optimizing the routability could have a poor reliability. This study presents a tradeoff between the two objectives through a hierarchical design. The method consists of three phases. First, cool and I/O cells are merged with hot cells to form clusters, thus the previous hypergraph is reduced to a shunk hypergraph. A quadisection partitioning algorithm then recursively divides the shunk hypergraph into two parts until every cluster is assigned to a placement area. Each hot cell is positioned on the center of its area. Second, a force-directed placement algorithm is used to position the cool cells subjected to the objective of wireability. Third, an interactive graphics package program is used to translate the relative placement to a real layout. Results indicate that our method can distribute the temperature profile on the substrate more uniformly and improves the system failure rate since both the maximum temperature and the maximum temperature variation are reduced. Following the reliability improvement the wireability only degrades slightly.