Математические вопросы кибернетики и вычислительной техники 20, 1998, 20–23.

Об одном методе минимизации площади БИС

И. А. Карапетян, С. Х. Дарбинян

Институт проблем информатики и автоматизации НАН РА и ЕрГУ
E-mail: isko@ipia.sci.am

Резюме

Задача, рассматриваемая в работе, возникает при сжатии эскиза топологии БИС или СБИС, полученной после трассировки. Предложен полиномиальный алгоритм порядка $O(n^{8/3})$ решения этой задачи.

Многие работы [1, 2, 3, 4] посвящены минимизации площади кристалла при стопроцентной трассировке соединений. Сущность этих работ заключается в нахождении путей деформации. Путь деформации [5] должен соединять противоположные края рисунка, не должен пересекать ни одной параллельной ему трассы и ни одного компонента (рис. 1.a). Если не удается найти такой путь, то ищется путь в виде ломаной (рис 1.$б$).

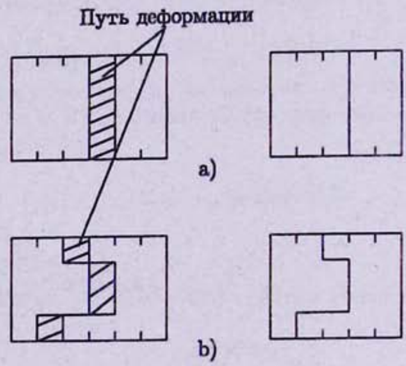

Рис.1

Найденный путь не должен пересекать компонентов и трасс, параллельных путям деформации. Затем сегменты найденных путей удаляются, и производится сжатие частей эскиза, как показано на рис.1 справа. Вышеописанные соображения приводят к следующей комбинаторной задаче [6]:

На числовой оси задан интервал $f = [a, b]$ и семейство его подинтервалов $F = \{f_1 = [a_1, b_1], f_2 = [a_2, b_2], \ldots f_n = [a_n, b_n]\}$. Требуется найти наибольшее число множеств

$A_1, A_2, ..., A_m$ из семейства подинтервалов F таких, что:

$$a) A_i \bigcap A_j = \emptyset, i, j = 1, 2, ..., m, i \neq j; \qquad (1)$$

$$b) \bigcup_{f_j \in A_i} f_j = f, i = 1, 2, .., m, \qquad (2)$$

т.е. объединение всех интервалов любого множества $A_i, i = 1, 2, ..., m$ есть интервал f.

Алгоритм решения этой задачи состоит из 3 этапов: построение сети, нахождение максимального потока в сети и определения множеств $A_1, A_2, ..., A_m$ из семейства F.

Построение сети. Сначала построим вспомогательную сеть $G' = (X', E')$ следующим образом. Каждому интервалу f_i из семейства F поставим в соответствие некоторую вершину $x_i, i = 1, 2, ..., n$. В качестве множества вершин X' сети $G' = (X', E')$ возьмем все вершины $x_i, i = 1, 2, ..., n$ и еще две другие вершины s и t, которые назовем, соответственно, источником и стоком сети. Множество дуг E' сети строим следующим образом. Источник s с помощью дуг, исходящих из него, соединяем со всеми вершинами x_i из множества X', левыми концами соответствующих интервалов которых является точка a, т.е. совпадает с левым концом интервала f. Аналогичным образом сток t с помощью дуг, входящих в него, соединяем со всеми вершинами x_j из множества вершин X', правыми концами соответствующих интервалов которых является точка b, т.е. совпадает с правым концом интервала f. Остальные дуги сети G' определяем следующим образом: вершины x_i и x_j соединяем дугой (x_i, x_j), если соответствующие их интервалы $f_i = [a_i, b_i]$ и $f_j = [a_j, b_j]$ пересекаются, ни один из этих интервалов полностью не содержится в другом и точка a_i (на числовой оси) находится левее, чем точка $a_j, i \neq j, i, j = 1, 2, ...n$.

Теперь, имея сеть $G' = (X', E')$ построим из нее сеть $G = (X, E)$ следующим образом. Обозначим через X'' подмножество всех вершин сети $G' = (X', E')$, у которых одновременно полустепень захода (число дуг, входящих в нее) и полустепень исхода (число дуг, исходящих из нее) больше единицы. При построении сети $G = (X, E)$ в сети $G' = (X', E')$ каждая вершина x_i из подмножества X'' заменяется двумя новыми вершинами x_i' и x_i'', причем каждая дуга (x, x_i) из сети $G' = (X', E')$ заменяется дугой (x, x_i') в сети $G = (X, E)$ и каждая дуга (x_i, x) из сети $G' = (X', E')$ заменяется дугой (x_i'', x) в сети $G = (X, E)$. Кроме того, добавим дугу (x_i', x_i'') между вершинами x_i' и x_i''. Будем считать, что пропускные способности всех дуг сети $G = (X, E)$ равны 1.

Максимальный поток. Допустим, имеем сеть $G = (X, E)$ с пропускными способностями дуг $q_{i,j} \geq 0$, источником s и стоком t; $s, t \in X$. Отметим, что множество чисел $a_{i,j} \geq 0$, определенных на дугах $(x_i, x_j) \in E$, называют потоком [7], если выполняются следующие условия:

$$\sum_{x_j \in \Gamma(x_i)} a_{i,j} - \sum_{x_k \in \Gamma^{-1}(x_i)} a_{k,j} = \begin{cases} v, \text{если} & x_i = s; \\ -v, \text{если} & x_i = t; \\ 0, \text{если} & x_i \neq s, t, \end{cases}$$

и $a_{i,j} \leq q_{i,j}$ для всех $(x_i, x_j) \in E$, где

$$\Gamma(x) = \{x'/x' \in X \quad \text{и} \quad (x, x') \in E\};$$

$$\Gamma^{-1}(x) = \{x'/x' \in X \quad \text{и} \quad (x', x) \in E\}.$$

В [7] доказано, что для нахождения максимального потока имеется полиномиальный алгоритм, являющийся модификацией алгоритма расстановки пометок. Там же, в частности, доказано, что время работы алгоритма в сетях с единичными пропускными способностями дуг не превосходит $O(|X|^{2/3}|A|)$, где A - списки смежностей, т.е. для каждой вершины $x \in X$ выписывается множество вершин $A(x) \subseteq X$, смежных с x.

Определение множеств $A_1, A_2, ..., A_m$. Допустим, что найден максимальный поток в сети $G = (X, E)$. Предположим, что все насыщенные дуги, входящие в сток t, каким-то образом пронумерованны начальными числами натурального ряда. Поочередно рассмотрим все насыщенные дуги, входящие в сток t. Для очередной насыщенной, скажем, i-ой (x, t) дуги поступаем следующим образом. Ищем насыщенную дугу (x', x), входящую в вершину x. Потом ищем насыщенную дугу, входящую в вершину x'. Этот процесс продолжаем до тех пор, пока не достигнем источника s. Из найденной цепи удалим вершины s и t. Для каждой вершины x_j оставшейся цепи берем соответствующий интервал f_j из семейства F, а для каждой пары вершин x'_k и x''_k, которые заменили вершину x_k при построении сети $G = (X, E)$, берем соответствующий интервал f_k из семейства F. Множество выбранных интервалов, соответствующих внутренным вершинам этой цепи, обозначим через A_i.

Теперь покажем, что найденный нами набор множеств $A_1, A_2, ..., A_m$ представляет собой решение вышесформулированной задачи. Действительно, согласно построению сети $G = (X, E)$, в каждую внутреннюю вершину, т.е. отличную от вершин s и t сети, самое большее входит только одна насыщенная дуга и выходит самое большее одна насыщенная дуга. Следовательно, никакая пара насыщенных цепей, начинающаяся в вершине s и кончающаяся в t , не имеет общих внутренних вершин. Поэтому $A_i \cap A_j = \emptyset$, при $i \neq j$ и $i, j = 1, 2, ...m$. То, что интервалы каждого множества $A_i, i = 1, 2, ...m$ покрывают интервал $f = [a, b]$, легко следует из построения сети $G = (X, E)$. В самом деле, двигаясь по любой насыщенной цепи , начиная от вершины s , легко видеть, что интервал, соответствующий первой встречающейся внутренней вершине этой цепи, имеет левый конец a. И интервал, соответствующий очередной внутренней вершине (паре внутрнних вершин) цепи, пересекается с интервалом, соответствущим предыдущей вершине (паре внутренних вершин) цепи. А интервал, соответствующий последней внутренней вершине цепи, имеет правый конец b. И, наконец, то, что количество найденных множеств, т.е. m - наибольшее, следует из максимальности потока.

Нетрудно заметить, что нахождение максимального потока - самый трудоемкий этап рассматриваемого алгоритма. Из построения сети $G = (X, E)$ следует, что число вершин сети не превосходит $2n + 2$. А для любой сети имеет место равенство $|A| = 2|E|$. Следовательно сложность вышеприведенного алгоритма не превосходит $O(|X|^{8/3})$.

Литература

[1] Г. Г. Рябов, Связность и преобразование дискретных конечных множеств: Труды семинара структурных и логических схем. ИТМ и ВТ АН СССР, Москва, 1969, с. 46–57.

[2] S. B. Akers, J. M. Geyer, D. L. Roberts, IC mask layout with a single conductor layer. Proc. 7th Design Automation Workshop, 1970, pp.7–16.

[3] A. E. Dunlop, SLIP: symbolic layout of integrated circuits with compaction. Computer Aided Design, November 1978, v.10, No 6, pp. 378–391.

[4] A. E. Dunlop, SLIM - the translation of symbolic layouts into mask date. Proc. 7th Design Automation Conf., 1980, pp.595-602.

[5] Л. Б. Абрайтис, Автоматизация проектирования топологии цифровых интегральных микросхем,"Сов. Радио", М., 1985, 200с.

[6] I. A. Karapetyan, On two Routing Problems, CSIT Conf., 1997, Yerevan, Armenia, Sept. 25-29, pp. 69–70.

[7] Х. Пападимитриу, К. Стайглиц, Комбинаторная оптимизация. Алгоритмы и сложность. "Мир", М., 1985, 512с.