УДК 681.325

АЛГОРИТМЫ ПОСЛЕДОВАТЕЛЬНОГО РАЗМЕЩЕНИЯ И КРАТЧАЙШЕЙ ТРАССИРОВКИ

Г.Д.Бочко, В.И.Кашин, Л.И.Макаров

Основные определения, используемые в данной работе, и постановка задач размещения элементов схемы в кристалле БИС и трассировки соединений схемы приведены в работе [1]. В этих задачах моделью схемы служит таблица связок $\|c_{ij}\|$, $i=\overline{1,n}$, $j=\overline{1,m}$, в которой для каждой из m связок (соединений) указываются элементы, входящие в нее (n - число элементов схемы), а моделью кристалла БИС служит плоская прямоугольная решетка $R=(A,U)$, $|A|\geq n$, имеющая множество A узлов, множество U ребер и заданную метрику с расстоянием $\rho_{\alpha\beta}$; $\alpha,\beta\in A$.

Алгоритм последовательного размещения предназначен для построения отображения элементов схемы в плоскую целочисленную прямоугольную решетку.

Макроблок-схема машинного алгоритма размещения приведена на рис.1.

Входной информацией алгоритма служат таблица связок и описание решетки, выходной информацией - описание соответствия элементов схемы узлам решетки (таблица размещения) и оценка суммарной длины трасс.

Алгоритм состоит из двух этапов:

А. Упорядочивание элементов схемы.

1. Построение матрицы смежности $\|d_{ik}\|$ по таблице связок $\|c_{ij}\|$; $i,k=\overline{1,n}$, $j=\overline{1,m}$,

$$d_{ik}=\sum_{j=1}^{m} c_{ij}c_{kj}$$

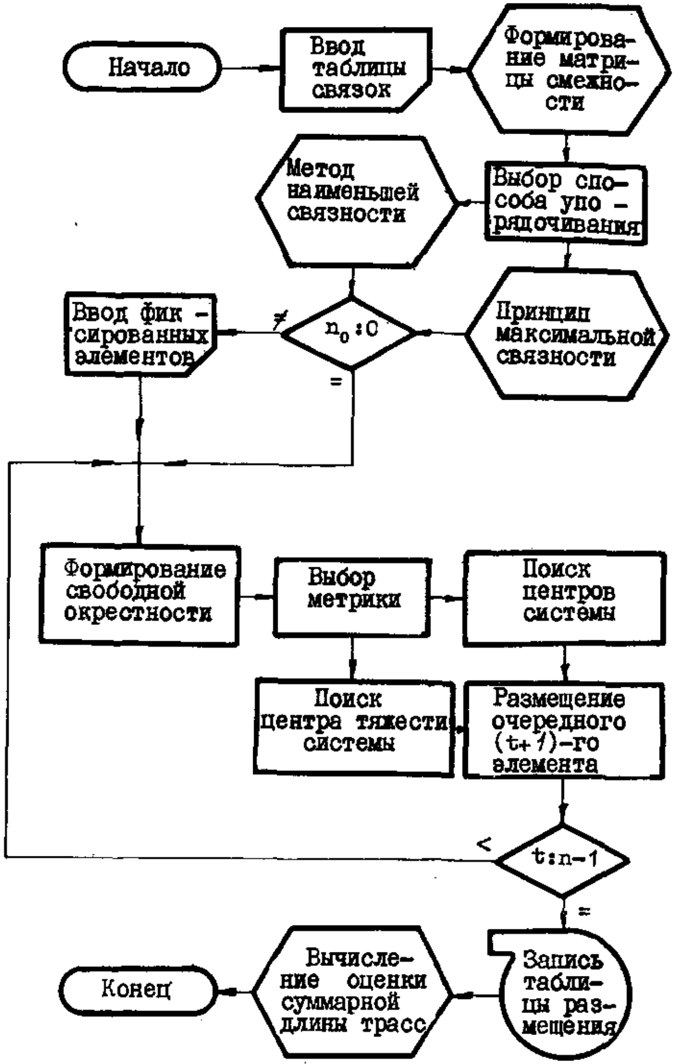

Рис.I. Алгоритм последовательного размещения.

2. Упорядочивание элементов схемы по матрице смежности.

При практическом использовании алгоритма для части элементов схемы может быть заранее задано отображение их в узлы решетки. Такие элементы схемы называют фиксированными и ставят их в начало упорядоченной последовательности элементов.

В алгоритме предусмотрено задание одного из двух способов упорядочивания.

а) Упорядочивание по принципу максимальной связности.

В качестве первого элемента упорядоченной последовательности, не содержащей фиксированных элементов, выбирается элемент, имеющий степень, равную $\max\limits_i d_i$, $d_i = \sum\limits_{k=1}^{n} d_{ik}$.

Далее, если упорядочено $(t-1)$ элементов, $2 \leq t \leq n$, то из остальных в качестве t-го элемента выбирается элемент, имеющий $\max\limits_i (2\delta_i - d_i)$, где δ_i – число ребер, соединяющих i-й неупорядоченный элемент со всеми упорядоченными.

б) Упорядочивание по методу наименьшей связности, предложенному Антиповой Т.И.

На s-м шаге упорядочивания ($s = 1, 2, \ldots, \left[\frac{n-n_0}{2}\right]$, n_0 – число фиксированных элементов) среди неупорядоченных элементов выбирается пара (i, k) элементов, для которой достигается $\min\limits_{i,k} (\Delta_i^{s-1} + \sigma_k^{s-1})$; i-у элементу в упорядоченной последовательности присваивается номер $t = (n - 2s + 1)$, а k-у – номер $(t+1)$, затем для каждого j-го неупорядоченного элемента находятся величины:

$$\Delta_j^{s} = \Delta_j^{s-1} - 2 d_{ij},$$

$$\sigma_j^{s} = \sigma_j^{s-1} - 2 d_{kj},$$

$$\Delta_j^{0} = \sigma_j^{0} = d_j,$$

которые используются на следующем шаге упорядочивания.

Б. Последовательное размещение.

В алгоритме предусмотрена возможность использования двух метрик решетки R :

– метрики с расстоянием $\rho = |x_i - x_k| + |y_i - y_k|$,

– метрики с расстоянием $\rho' = [(x_i - x_k)^2 + (y_i - y_k)^2]^{1/2}$,

где (x_i, y_i) и (x_k, y_k) – координаты i-го и k-го узлов решетки.

Пусть в решетке $R=(A,U)$ задана система узлов $V\subset A$. В качестве оценки длины трассы, соединяющей узлы системы $V, |V|=t$, в зависимости от выбранной метрики используются величины

$$\ell_0=\sum_{i\in V}\rho_{ic},\quad c\in C,$$

$$\ell_0'=\sum_{i\in V}\rho_{ib}',$$

где C – множество центров системы V, b – центр тяжести системы V, т.е.

$$x_b=\frac{1}{t}\sum_{i\in V}x_i,\quad y_b=\frac{1}{t}\sum_{i\in V}y_i.$$

Для заданных решетки R и системы V часть плоскости, для множества точек которой выполняются соотношения

$$x_0=\min_{i\in V}x_i\leq x\leq\max_{i\in V}x_i=x_0',$$

$$y_0=\min_{i\in V}y_i\leq y\leq\max_{i\in V}y_i=y_0',$$

назовем прямоугольником P, а часть решетки R, для множества $\{k\}$ узлов которой выполняются

$$x_0-1\leq x_k\leq x_0'+1,$$

$$y_0-1\leq y_k\leq y_0'+1,$$

назовем решеткой R^0. Тогда справедлива

ТЕОРЕМА I. Множество центров C и центр тяжести b системы V принадлежат прямоугольнику P, а ближайшие к ним узлы решетки R принадлежат решетке R^0.

Итак, этап последовательного размещения состоит в следующем.

I. Размещение фиксированных элементов.

2. Размещение очередного ($t+1$)-го элемента, $0\leq t\leq n-1$.

По матрице смежности находится система V_t, размещенных элементов, связанных с ($t+1$)-м, прямоугольник P_t и решетка R_t^0. В прямоугольнике P_t находится множество C центров системы V_t (или центр тяжести b). В решетке R_t^0 или в решетке R находится свободный узел s, для которого достигается $\min_{c\in C}\min_{s\in A\setminus V}\rho_{sc}$ (или $\min_{s\in A\setminus V}\rho_{sb}$), V – множество узлов, в которых размещены t предыдущих элементов.

В узел s помещается ($t+1$)-й элемент.

Если $t=n-1$, то размещение закончено, т.е. сформирована таблица размещения.

3. Вычисление оценки суммарной длины трасс.

Объем машинной памяти и время счета, необходимые для размещения в решетке схемы из n элементов, содержащей m связок, пропорциональны nm.

Алгоритм кратчайшей трассировки в плоской прямоугольной решетке предназначен для построения трассы, соединяющей узлы решетки, соответствующие элементам одной связки [I,2].

Макроблок-схема машинного алгоритма трассировки приведена на рис.2.

Входной информацией алгоритма служат таблица связок и таблица размещения элементов схемы в решетке, выходной информацией – описание трассы для каждой связки и значение ее длины.

I. При построении трассы T_j очередной j-й, $j=\overline{1,m}$, связки из таблицы размещения выделяется система V_j, $|V_j|=n_j$, узлов, в которых размещены элементы данной связки.

П. Для нахождения кратчайшего дерева $D_j=(V_j,U_j')$, покрывающего полный взвешенный граф $\Gamma_j=(V_j,U_{V_j})$, $U_j'\subseteq U_{V_j}$, соответствующий системе V_j, применяется модификация алгоритма Прима [3], основанного на принципе "ближайшего соседа".

I^0. Выбирается начальная (стартовая) вершина; считаем ее построенной частью (фрагментом) дерева D_j. Стартовая вершина может быть задана либо произвольно, либо все вершины поочередно выбираются в качестве стартовых.

2^0. К построенному фрагменту, содержащему V_0 вершин, присоединяется ближайшая вершина α и соответствующее ребро (α, i), т.е. вершина, для которой $\rho_0=\rho_{\alpha i}=\min_{\alpha'}\min_{i'}\rho_{\alpha' i'}$, $\alpha'\in V_j\setminus V_0$, $i'\in V_0$, причем

а) если $\rho_0=\rho_{\alpha_1 i}=\rho_{\alpha_2 i}$ и $\alpha_1<\alpha_2$; $\alpha_1,\alpha_2\in V_j\setminus V_0$, то $\alpha=\alpha_1$, т.е. из нескольких ближайших к i-й, $i\in V_0$, вершин присоединяется вершина с меньшим номером;

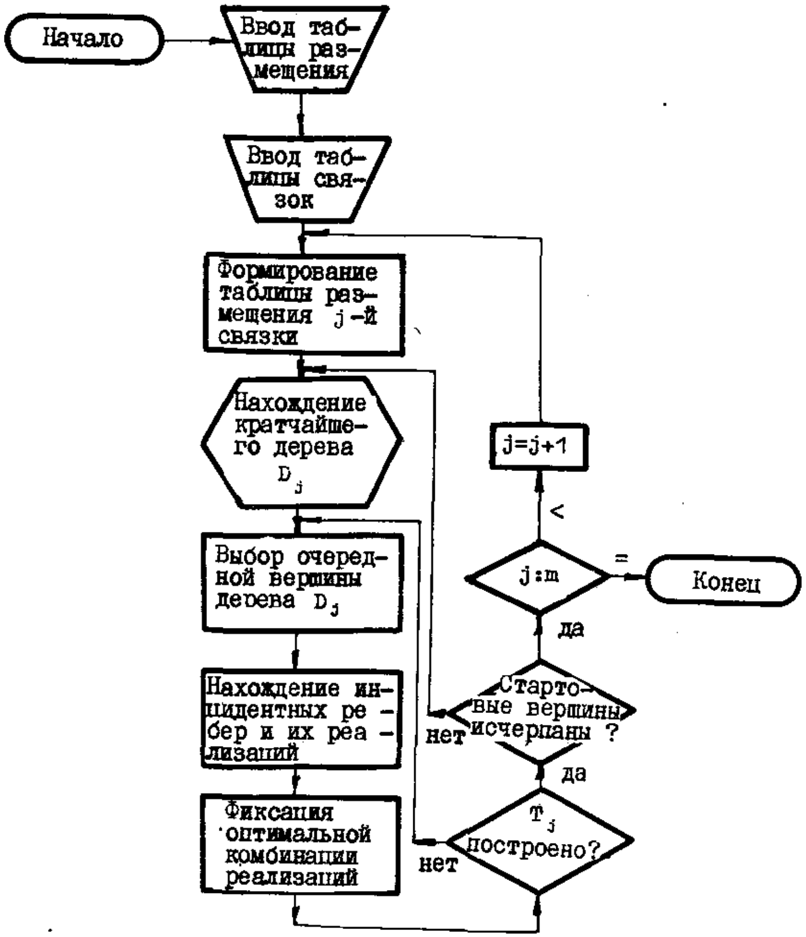

Рис.2. Алгоритм кратчайшей трассировки

б) если $\rho_0=\rho_{\alpha_1 i_1}=\rho_{\alpha_2 i_2}$ и $i_1<i_2$; $i_1, i_2 \in V_0$, то $i=i_1$, $\alpha=\alpha_1$, т.е. к фрагменту присоединяется вершина, смежная вершине фрагмента, имеющей меньший номер.

3°. Если $|V_0|=n_j$, то кратчайшее дерево $\mathcal{D}_j$ построено; иначе переход к п. П.2°.

Поскольку расстояния между парами вершин могут быть вычислены по таблице размещения элементов одной связки, то объём машинной памяти и время счета, необходимые для построения кратчайшего дерева, пропорциональны n_j и n_j^2, соответственно.

Ш. Совмещение реализаций ребер кратчайшего дерева.

Реализациями в решетке ребра (i,k) кратчайшего дерева назовем два кратчайших пути ℓ_1 и ℓ_2, соединяющих в решетке узлы (x_i, y_i) и (x_k, y_k) и проходящих через узлы (x_i, y_k) и (x_k, y_i), соответственно.

I°. Из описания кратчайшего дерева $\mathcal{D}_j$ выбирается (начиная со стартовой) очередная вершина.

2°. Для очередной вершины находятся в дереве $\mathcal{D}_j$ все инцидентные ей ребра (число ребер не превышает 8) и их реализации (ℓ_1 и ℓ_2) в решетке. Реализации ребер, зафиксированные для предыдущих вершин, отмечаются.

3°. Выбирается (с учетом отмеченных) такая комбинация найденных реализаций, которая имеет наибольшее совмещение и содержит для каждого ребра единственную (ℓ_1 или ℓ_2) реализацию. Выбранная комбинация фиксируется.

4°. Если все ребра дерева $\mathcal{D}_j$ реализованы в решетке, то трасса T_j построена; иначе переход к п. Ш.2°.

IУ. Если $j=m$, то конец; иначе переход к п.I.

Результатом работы п.Ш алгоритма является описание трассы T_j, соединяющей узлы решетки, соответствующие j-й связке. Для любой трассы T, построенной данным алгоритмом, справедлива

ТЕОРЕМА 2. Трасса T является деревом.

Реализации неинцидентных ребер дерева $\mathcal{D}$ назовем пересекающимися, если они содержат общие узлы или ребра решетки. Трасса T, не содержащая пересекающихся реализаций, является деревом, так как в этом случае каждой цепи дерева $\mathcal{D}$ соответствует единственная часть трассы T, состоящая из реализаций ребер этой цепи. Пусть трасса T содержит пересекающиеся реализации ребер (a, b) и (c, d) и часть T' трассы, соответствующую цепи дерева $\mathcal{D}$, соединяющей вершины b и c и не содержащей вершины a и d (рис.3). Пусть вершина a присоединена к дереву $\mathcal{D}$ при его построении ранее вершин b, c, d (для других случаев доказательство аналогично). Тогда последовательность присоединения вершин к дереву $\mathcal{D}$ следующая: a, b, c, d (иначе в дереве $\mathcal{D}$ содержались бы две различные цепи, соединяющие вершины a и d). Отсюда и из принципа "ближайшего соседа" следует $\rho_{ab} \le \rho_{ac}$, т.е. $y \le x$, $\rho_{cd} \le \rho_{bd}$, т.е. $x \le y$. Отсюда получаем равенства: $x = y$, $\rho_{ab} = \rho_{ac}$, $\rho_{cd} = \rho_{bd}$.

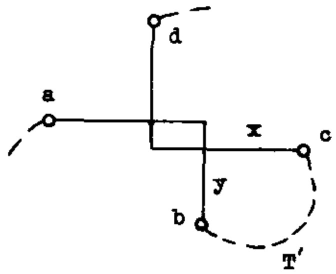

Рис. 3

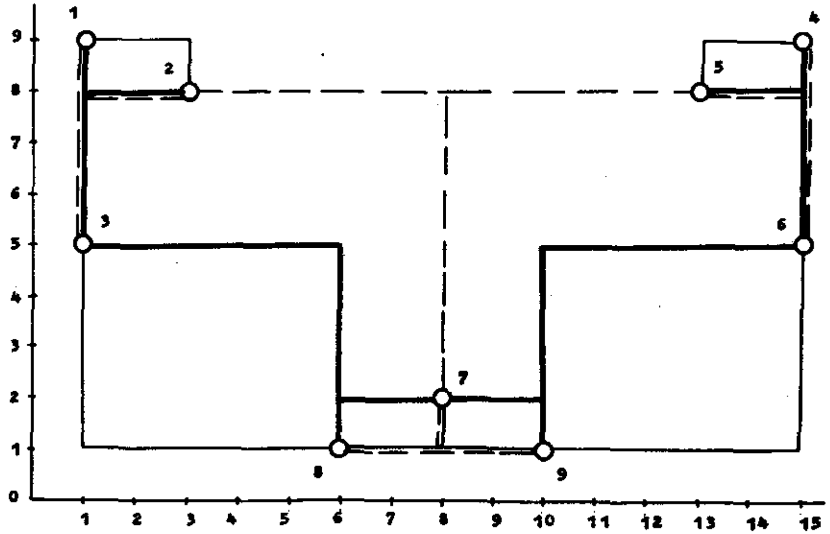

Рис. 4

Из найденных равенств и условия П.2°,а) алгоритма следует, что $b < c$ и при выполнении условия П.2°,б) ребро (c, d) не может входить в дерево $\mathcal{D}$ (вместо него в дерево $\mathcal{D}$ входит ребро (b, d)). Таким образом, трасса T не содержит пересекающихся реализаций, т.е. является деревом.

Поскольку число ребер, инцидентных каждой вершине дерева $\mathcal{D}_j$, не превышает константы, то объём машинной памяти и время счета, необходимые для построения трассы T_j, пропорциональны n_j и n_j^2, соответственно. Отсюда следует, что объем машинной памяти и время счета, необходимые для трассировки всех m связок схемы, пропорциональны nm и $\sum_{j=1}^{m} n_j^2$, соответственно.

Пример трассы, построенной алгоритмом, приведен на рис.4 (жирная линия). Тонкими линиями показаны пути, не использованные при построении трассы. Пунктиром показано кратчайшее дерево Штейнера, имеющее длину равную 33. Длина кратчайшего дерева $\mathcal{D}$ равна 38, а длина трассы – 34.

Приведенные алгоритмы последовательного размещения и кратчайшей трассировки реализованы в программах, написанных на языке ФОРТРАН для ЭВМ "Минск-32", при этом общий объем программ составляет 3348 машинных команд. Программы позволяют размещать схемы с числом элементов $n \le 1000$ и проводить трассировку соединений для связок с числом элементов $n_j \le 500$, $j = \overline{1, m}$.

Литература

1. МАКАРОВ Л.И. Размещение и трассировка в плоской прямоугольной решетке. –Настоящий сборник, с.65–72.

2. БОЧКО Г.Д., МАКАРОВ Л.И. Один алгоритм трассировки в прямоугольной решетке. –В кн.: Вычислительная техника. Материалы конф. по развитию технических наук в республике и использованию их результатов и межотраслевого республиканского совещания по применению ЭВМ для проектирования средств вычислительной техники. Каунас, 1974, с.111–113. (Каунасский политехнический институт).

3. ПРИМ Р.К. Кратчайшие связывающие сети и некоторые обобщения.–"Кибернетический сб.", вып. 2, 1961, с. 95–107.

Поступила в ред.-изд.отд.
3 апреля 1975 года