Кроме временных задержек, в настоящее время, все более важными становятся критерии минимизации тепловыделения, потребления мощности, функциональной полноты СБИС. Поэтому необходимы дальнейшее исследования и разработки в области многокритериальных алгоритмов разбиения с использованием эффективных методов поиска квазиоптимальных решений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Афанасьев И.В., Венгер О.В., Марченко А.М.* Применение кластеризации при логическом синтезе//II Всероссийская научно-техническая конференция «Проблемы разработки перспективных микроэлектронных систем - 2006»/МЭС-2006, стр. 40-44
2. *Lawler, E. L., K. N. Levitt and j. Turner.* "Module Clustering to Minimize Delay in Digital Networks", IEEE Transactions on Computers. Vol C-18 No. 1, 1966, p. 47-77
3. *Murgai, R., R.K. Brayton and A. Songiovanni-Vincentelli.* "On Clustering for Minimun Delay/Area", Proc. of the IEEE Int'l Conf. On Computer-Aided Design, Nov., 1991, p. 6-9
4. *J. Cong, H. Li, and C. Wu.* Simultaneous Circuit Partitioning/Clustering with Retiming for Performance Optimization, In Proc. ACM/IEEE Design Automation Conference, p. 460–465, 1999.
5. *R. Aggarwal, R. Murgai, and M. Fujita.* Speeding Up Technology-Independent Timing Optimization by Network Partitioning // Proc. ACM/IEEE Design Automation Conference. - Nov. 1997. - p. 83-90.
6. *D. Banieres, J. Cortadella, and M. Kishinevsky.* Dominatorbased Partitioning for Delay Optimization, Great Lake Symposium on VLSI, 2006.
7. *S. Dey, F. Brglez, G. Kedem.* Circuit Partitioning for Logic Synthesis // IEEE Journal of Solid-state Circuits. – March 1991. - V. 26. - № 3.
8. *Y. Nakamura and T. Yoshimura.* A Partitioning-based Logic Optimization Method for Large Scale Circuits with Boolean Matrix // Proc. ACM/IEEE Design Automation Conference, 1995.
9. *G. Karypis, R. Aggrwal, V. Kumar, and S. Shekhar.* Multilevel hypergraph partitioning: Application in VLSI domain // Proc. ACM/IEEE Design Automation Conference, 1997.
10. *Elmore W.C.* The Transient Response of Damped Linear Networks with Particular Regards to Wide-Band Amplifies // J. Appl. Phys. - 1948. - V. 19. - P. 55–63.
11. *Cong J.J., Leung K.-S.* Optimal Wiresizing Under Elmore Delay Model // IEEE Trans. on CAD of Integrated Systems. - 1995. - V. 14. - № 3. - P. 321–336.
12. *Chen C.P., Chen Y.P., Wong D.F.* Optimal Wiresizing Under Elmore Delay Model // IEEE Trans. on CAD of Integrated Systems. - 2002. - V. 21. - No. 3. - P. 319–329.

М.А. Бакало, В.В. Курейчик

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ РАЗМЕЩЕНИЯ МЕТОДОМ ПАРНЫХ ПЕРЕСТАНОВОК*

Введение. Одним из основных этапов конструкторского проектирования ЭВА является размещение элементов (модулей) на некотором коммутационном поле с оптимизацией заданных наперед критериев качества. Среди существующих алгоритмов отдельную группу составляют итерационные алгоритмы размещения, к которым и относится алгоритм парных перестановок [1-3].

Задача размещения может быть сформулирована следующим образом: дано множество элементов $V = \{e_i \mid i = \overline{1,n}\}$, соединенных друг с другом множеством связей $U \subseteq V \times V$. В качестве модели представления данных используем неори-

* Работа выполнена при частичной финансовой поддержке программы развития научного потенциала высшей школы 2006-2008 годы (проекты РНП.2.1.2.2238, РНП 2.1.2.3193).

ентированный граф $G(V,U)$, при этом информация о связях записывается я в матрицу смежности, которая имеет вид: $R=\left\|r_{i,j}\right\|$, где $i, j=\overline{1,n}$. Под областью размещения будем понимать множество позиций для размещения элементов, организованные в прямоугольную матрицу, имеющие одинаковые линейные размеры и характеризующиеся их координатами. При этом позиции, объединенные в строки и столбцы, имеют одинаковые X- и Y-координаты соответственно.

Необходимо построить размещение $P=\langle p_1, p_2,..., p_n\rangle$ такое, что каждой позиции области размещения p_i соответствует некоторый элемент $e_i \in V$ при этом суммарная длина связей элементов:

$$F=\sum_{i=1}^{n}\sum_{j=1}^{n} r_{i,j}\cdot d_{i,j} \to \min. \quad (1)$$

В алгоритме размещения, описанным в работе Глазера [2] упорядочивание элементов $e_i \in V$ происходит в соответствии с убыванием и рассчитывается по формуле:

$$L_i=\sum_{j=1}^{n} r_{i,j}\cdot d_{i,j}, \quad (2)$$

$i=\overline{1,n}$, $r_{i,j}$ – вес ребра в матрице смежности $R=\left\|r_{i,j}\right\|$, а $d_{i,j}$ – длина связи элементов e_i и e_j. Получим последовательность элементов: $E=\langle e_1,e_2,...,e_n\rangle$.

Вычислительная сложность этого алгоритма размещения пропорционально $O(n^4)$ [2,4].

1. Критический анализ алгоритма размещения

1.1. Анализ формулы вычисления приращения суммарной длины связей. Проанализируем формулы (1) и (2) оценки приращения суммарной длины связей размещения.

По отношению к элементам e_k и e_t все множество элементов (вершин графа) разбиваются на следующие подмножества (рис.1): S_k – множество элементов, связанных только с элементом e_k; S_t – множество элементов, связанных только с элементом e_t; $S_{k,t}$ – множество элементов связанных как с элементом e_k, так и с элементом e_t; S – множество элементов не связанных ни с элементом e_k, ни с элементом e_t.

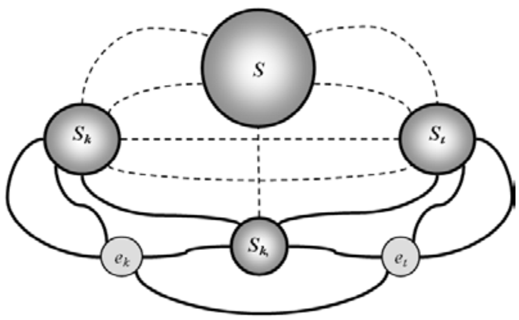

Рис.1. Подмножества элементов, выделенные относительно элементов e_k и e_t

Проведем оценку величины, которую каждое из этих подмножеств вносит в суммарную длину связей всего размещения: Суммарная длина связей элемента e_k со всеми элементами подмножества S_k: $D_k = \sum_{l \in S_k} r_{k,l} \cdot d_{k,l}$.Суммарная длина связей элемента e_t со всеми элементами подмножества S_t: $D_t = \sum_{l \in S_t} r_{t,l} \cdot d_{t,l}$.Суммарная длина связей элементов e_k и e_t с подмножеством $S_{k,t}$: $DD_{k,t} = DD_k + DD_t$, где DD_k – суммарная длина связей элемента e_k с подмножеством $S_{k,t}$: $DD_k = \sum_{l \in S_{k,t}} r_{k,l} \cdot d_{k,l}$, а DD_t – суммарная длина связей элемента e_t с подмножеством $S_{k,t}$: $DD_t = \sum_{l \in S_{k,t}} r_{t,l} \cdot d_{t,l}$. Суммарная длина связей всех элементов, не смежных ни с элементом e_k, ни с элементом e_t: $DS = \sum_{\substack{i=1 \\ i \neq k, i \neq t, j \neq k, j \neq t}}^{n} \sum_{j=1}^{n} r_{i,j} \cdot d_{i,j}$. С учетом положений, формула (1) вычисления суммарной длины связей преобразуется к виду:

$$F = D_k + D_t + DD_{k,t} + DS . \tag{5}$$

С учетом формул (11) величину $\Delta F_{k,t}$ можно записать в следующем виде:

$$\begin{aligned} \Delta F_{k,t} &= (D_k - D_k^*) + (D_t - D_t^*) + (DD_{k,t} - DD_{k,t}^*) + (DS_{k,t} - DS_{k,t}^*) = \\ &= (D_k - D_k^*) + (D_t - D_t^*) + (DD_k - DD_k^*) + (DD_t - DD_t^*), \end{aligned} \tag{6}$$

где D^*_k, D^*_t, $DD^*_{k,t}$, DS^* – величины, рассчитанные по приведенным выше формулам после перестановки местами элементов e_k и e_t.

Очевидно, что величины DS и DS^* совпадают, поскольку они учитывают суммарную длину связей элементов, не смежных с элементами e_k и e_t, соответственно перестановка этих элементов не модифицирует указанные связи (помечены пунктирными линиями (см. рис.1)), соответственно величина $(DS_{k,t} - DS_{k,t}^*) = 0$.

Введя обозначение ΔD_k для изменения суммарной длины связей элемента e_k после перестановки: $\Delta D_k = (D_k + DD_k) - (D_k^* + DD_k^*)$, и ΔD_t – для изменения суммарной длины связей элемента e_t после перестановки: $\Delta D_t = (D_t + DD_t) - (D_t^* + DD_t^*)$, преобразуем формулу (6) к следующему виду:

$$\Delta F_{k,t} = \Delta D_k + \Delta D_t . \tag{7}$$

Таким образом, приращение суммарной длины связей размещения $\Delta F_{k,t}$, выступающее критерием оптимизации и рассчитываемое на шаге 4 алгоритма, сводится к приращению суммарной длины связей элементов e_k и e_t после перестановки этих элементов местами и не зависит от суммарной длины связей элементов, не участвующих в перестановке.

Следствием данного факта является существенное упрощение алгоритма. В итоге, применение формулы (7) позволяет уменьшить временную сложность 4 шага алгоритма с величины $O(n^2)$ до $O(n)$.

1.2. Анализ формулы вычисления приращения суммарной длины связей элементов e_k и e_t. Проанализируем приращение суммарной длины связей элемента

e_k при установке его в позицию элемента e_t. В дальнейших рассуждениях мы будем придерживаться следующих обозначений: (x_k, y_k) – координаты элемента e_k до перестановки; (x_t, y_t) – координаты элемента e_k после перестановки (соответствуют координатам элемента e_t до перестановки); (x_l, y_l) – координаты элемента $e_l \in (S_k \cup S_{k,t})$.

При проектирования ЭВА соединения между элементами схемы в большинстве случаев трассируются под прямым углом. Поэтому для оценки длины соединения элементов e_k и e_l будем использовать не Евклидово расстояние, а длину дерева Штейнера, длина ребра которого рассчитывается по формуле:

$$d_{k,l} = |x_k - x_l| + |y_k - y_l|. \tag{8}$$

С учетом формулы (8) суммарная длина связей элемента e_k будет рассчитываться по следующей формуле:

$$D_k + DD_k = \sum_{e_l \in (S_k \cup S_{k,t})} r_{k,l} \cdot \left(|x_k - x_l| + |y_k - y_l|\right). \tag{9}$$

Суммарная длина связей элемента e_k после перестановки будет рассчитываться по формуле (9) с учетом того, что новые координаты элемента e_k после перестановки совпадают с координатами элемента e_t до перестановки. Формула (9) позволяет нам уточнить формулу (6) для расчета приращения суммарной длины связей элемента e_k, связанного с перестановкой элемента из позиции с координатами (x_k, y_k) в позицию с координатами (x_t, y_t):

$$\Delta D_k = \sum_{e_l \in (S_k \cup S_{k,t})} r_{k,l} \cdot \left[\left(|x_k - x_l| - |x_t - x_l|\right) + \left(|y_k - y_l| - |y_t - y_l|\right)\right]. \tag{10}$$

Подавляющее большинство итерационных алгоритмов для просмотра вершин используют линейную последовательность. Как будет доказано ниже, такой подход не является оптимальным, поскольку в ходе вычисления суммарной длины связей элемента e_k некоторые величины, оставаясь постоянными, вычисляются многократно.

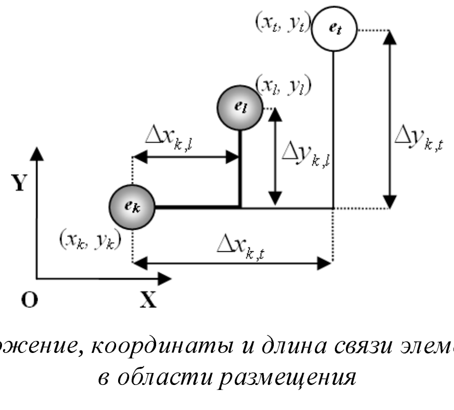

Рис.2. Расположение, координаты и длина связи элементов e_k, e_t и e_l в области размещения

Докажем приведенное выше утверждение, при этом формула (10) будет иметь другой вид. В зависимости от взаимного расположения элементов e_k и e_t область размещения может быть разделена на девять подобластей (рис.3).

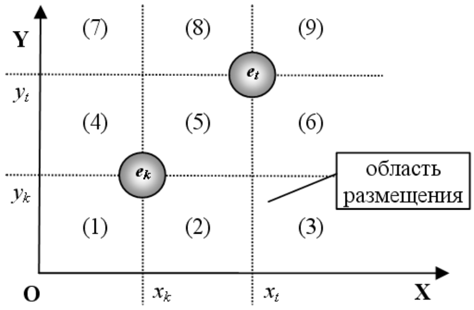

Рис.3. Разбиение области размещения на подобласти (1) – (9) в зависимости от расположения элементов e_k и e_t, соответствующего случаю (a)

Возможны четыре случая взаимного расположения элементов e_k и e_t:
a) элемент e_t расположен в правом верхнем квадранте относительно элемента e_k;
b) элемент e_t расположен в левом верхнем квадранте относительно элемента e_k;
c) элемент e_t расположен в левом нижнем квадранте относительно элемента e_k;
d) элемент e_t расположен в правом нижнем квадранте относительно элемента e_k;

Независимо от взаимного расположения элементов e_k и e_t будем придерживаться приведенного выше порядка нумерации подобластей.

Как можно видеть, X-координаты элементов e_k и e_t разбивают ось Абсцисс на три интервала, в один из которых может попадать X-координата элемента e_l. Формулы для расчета приращения длины связи элементов e_k и e_l вдоль оси абсцисс, в зависимости от взаимного положения элементов e_k, e_t и e_l имеют вид:

Интервал	***Формула приращения:***
$\mathbf{x_l} < x_k < x_t$ $\mathbf{x_l} < x_t < x_k$	$\Delta x = (x_k - x_l) - (x_t - x_l) = (x_k - x_t)$
$x_k < \mathbf{x_l} < x_t$	$\Delta x = -(x_k - x_l) - (x_t - x_l) == 2x_l - (x_k + x_t)$
$x_t < \mathbf{x_l} < x_k$	$\Delta x = (x_k - x_l) + (x_t - x_l) = -2x_l + (x_k + x_t)$
$x_k < x_t < \mathbf{x_l}$ $x_t < x_k < \mathbf{x_l}$	$\Delta x = -(x_k - x_l) + (x_t - x_l) = -(x_k - x_t)$

Приведенные выше формулы свидетельствуют, что для всех элементов, расположенных в областях (1), (3), (4), (6), (7) и (9), величина приращения вдоль оси абсцисс не зависит от координат элемента e_l, а определяется разницей координат $\Delta x_{k,t} = (x_k - x_t)$, которая является постоянной для заданных e_k и e_t. Для элементов, расположенных в областях (2), (5) и (8), приращение учитывает координат элемента e_l, однако, содержит величину $(x_k + x_t)$, которая также остается постоянной для заданных e_k и e_t.

Аналогичные рассуждения позволяют вывести формулы для расчета приращения длины связи элементов e_k и e_l вдоль оси ординат, в зависимости от взаимного положения элементов e_k, e_t и e_l:

Интервал	*Формула приращения:*
$y_l < y_k < y_t$ $y_l < y_t < y_k$	$\Delta y = (y_k - y_l) - (y_t - y_l) = (y_k - y_t)$
$y_k < y_l < y_t$	$\Delta y = -(y_k - y_l) - (y_t - y_l) = 2y_l + (y_k + y_t)$
$y_t < y_l < y_k$	$\Delta y = (y_k - y_l) + (y_t - y_l) = -2y_l + (y_k + y_t)$
$y_k < y_t < y_l$ $y_t < y_k < y_l$	$\Delta y = -(y_k - y_l) + (y_t - y_l) = -(y_k - y_t)$

Для приращения вдоль оси ординат постоянными для заданных e_k и e_t, остаются величины $(y_k - y_t)$ и $(y_k + y_t)$.

Логическим продолжением рассуждений, приведенных выше, является переход к использованию при вычислении приращения суммарной длины связей элемента e_k вместо одномерной последовательности элементов – двумерной последовательности. При таком подходе, вычисление частичного приращения, которое вносят элементы каждой из подобластей (1) – (9), позволит избежать многократного вычисления постоянных величин, которое неизбежно при последовательном переборе элементов. Иначе говоря, при вычислении суммарной длины связей будем перебирать не элементы e_i, а позиции области размещения. Поскольку число позиций области размещения соответствует числу элементов, и между позицией и элементов существует взаимнооднозначное соответствие, временная сложность вычисления приращения суммарной длины связей при перестановке элементов e_k и e_t, будет оцениваться величиной $O(n)$.

Введем следующие обозначения: I_{max} – число позиций области размещения по горизонтали; J_{max} – число позиций области размещения по вертикали; (i, j) – индексы позиции области размещения; $p_{i,j}$ – элемент e_l, соответствующий позиции с индексами (i, j), соответственно вес связи некоторого элемента e_k с элементом e_l, заданным позицией $p_{i,j}$ будем обозначать как $r_{k,l}$; (I_k, J_k) – индексы позиции элемента e_k; (I_t, J_t) – индексы позиции элемента e_t, которые соответствуют позиции элемента e_k после перестановки.

В данных обозначениях мы можем выразить частичные приращение суммарной длины связей элемента e_k с элементами каждой из подобластей (1) – (9), вызванное его перестановкой в позицию элемента e_t. При этом будем иметь ввиду, что постоянные величины можно вынести из под знака суммы.

для подобласти (1):

$$\Delta D_k^{(1)} = \sum_{i=1}^{I_k} \sum_{j=1}^{J_k} r_{k,l} \cdot \left((x_k - x_t) + (y_k - y_t) \right) = \left((x_k - x_t) + (y_k - y_t) \right) \cdot \sum_{i=1}^{I_k} \sum_{j=1}^{J_k} r_{k,l} \,. \quad (11)$$

Аналогично вычисляются частичные приращение суммарной длины связей элемента e_k с элементами каждой из других подобластей. Обобщенное выражение для расчета приращения суммарной длины связей элемента e_k, вызванное его перестановкой в позицию элемента e_t при их взаимном расположении, соответствующем случаю (a), приобретет вид:

$$\Delta D_k^{(a)} = \Delta D_k^{(1)} + \Delta D_k^{(2)} + \Delta D_k^{(3)} + \Delta D_k^{(4)} + \Delta D_k^{(5)} + \Delta D_k^{(6)} + \Delta D_k^{(7)} + \Delta D_k^{(8)} + \Delta D_k^{(9)} \,. \quad (12)$$

Аналогичные рассуждения позволяют вывести формулы расчета частичного приращения суммарной длины связей с элементами каждой из подобластей (1) – (9), для взаимного расположения элементов e_k и e_t, соответствующего случаям (*b*), (*c*), (*d*).

2. Описание модифицированного алгоритма. Обратим внимание, что изменения касаются шага 4) исходного алгоритма.

1) Формирование начального размещения.
2) Упорядочивание элементов e_i в соответствии с убывание характеристики, рассчитываемой по формуле (1). $E = \langle e_1, e_2, \ldots, e_n \rangle$.
3) Выбираем элемент e_k, $k = 1$.
4) Определяем позицию p_k области размещения и индексы позиции (I_k, J_k) и координаты (x_k, y_k), соответствующие элементу e_k.
5) Выбираем элементов e_t, $t = \overline{k+1, n}$.
6) Определяем позицию p_t области размещения и индексы позиции (I_t, J_t) и координаты (x_t, y_t), соответствующие элементу e_t.
7) На основании данных, полученных при выполнении шагов 4) и 6) определяем взаимное расположение элементов e_k и e_t в области размещения – случаи a), b), c) или d) и, соответственно, выбираем формулы для расчета частичного суммарной длины связей элементов e_k и e_t с элементами каждой из подобластей (1) – (9), вызванное их перестановкой.
8) По формуле (7) взаимного расположения элементов e_k и e_t вычисляем приращение суммарной длины связей $\Delta F_{k,t}$.
9) Выбираем следующий элемент e_t, $t := t + 1$. Если $t \le n$, то переходим к шагу 6.
10) Если для элемента e_k найдено положительное приращение $\Delta F_{k,t}$, то производим перестановку элементов e_k и e_t, для которого такое приращение было максимально.
11) Выбираем следующий элемент e_k для уточнения его местоположения $k := k + 1$. Если $k \le n$, то переходим к шагу 4.
12) Если не выполнено условия остановки, то переходим к шагу 3.
13) Например, правило остановки может быть следующим: улучшение размещения ΔF на *i-й* итерации составило величину, меньшую заранее заданного значения $\Delta F_{\lim}$: $\Delta F = F(P_{i-1}) - F(P_i) \le \Delta F_{\lim}$.
14) Конец.

Вычислительная сложность модифицированного алгоритма $O(n^2)$.

3. Экспериментальные исследования. В целях проведения объективной оценки эффективности предложенного метода оптимизации размещения, были проведены серии испытаний. Для сравнения были реализованы: алгоритм, реализующий случайное размещение; алгоритм оптимизации размещения методом парных перестановок (Глазера) и разработанный модифицированный алгоритм.

Кроме того, производилось сравнение эффективности указанных алгоритмов на областях размещения различной конфигурации.

Отображенные на графике (рис.4) результаты были получены следующим образом: генерировалось несколько графов (от 10 до 30) с одинаковым числом вершин, но различающихся числом ребер. Следует отметить, что оценка размещения, полученная после оптимизации случайного размещения, для всех использованных методов совпадает, поэтому для иллюстрации эффективности оптимизации приведем только график отношения величины приращения к величине суммарной длины связей.

Рис.4. Результаты исследования алгоритмов размещения

Заключение. Проведенные исследования позволили модифицировать алгоритм размещения и уменьшить его вычислительную сложность до величины $O(n^2)$. Предложенный подход к вычислению приращения суммарной длины связей как суммы частичных приращений, вносимых различными подобластями области размещения, также может быть применен для алгоритмов групповой перестановки, а также для оптимизации размещения в 3D-области.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Курейчик В.М.* Математическое обеспечение конструкторского и технологического проектирования с применением САПР. – М.: Радио и связь, 1990.
2. *Кормен Т., Лейзерсон И., Ривест Р.* Алгоритмы: построения и анализ. – М.: МЦМО, 2000.
3. Sherwani Naveed. Algorithms for VLSI Physical Design Automation, Kluwer Academic Publishers, Boston/Dordrecht/London, 1995.
4. *Г Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982.

В.М. Глушань, Р.В. Иванько, Р.М. Романов, М.Д. Сеченов

ПРАВОВЫЕ АСПЕКТЫ ПОСТРОЕНИЯ И ИСПОЛЬЗОВАНИЯ РАСПРЕДЕЛЕННЫХ САПР

Введение. Концепция распределенных вычислительных систем (РВС) появилась около четверти века назад и базируется она на понятии процессорной группы, впервые введенного разработчиками экспериментальной распределенной операционной системы MEDUSA в 1980 году [1]. Под этим термином понимается множество процессов, взаимодействующих для согласованного решения общей задачи. Справедливости ради следует заметить, что концепция параллельных вычислений на многопроцессорных вычислительных структурах (МВС) появилась гораздо раньше, а для САПР она воплотилась в специализированные моделирующие