УДК 881.3.06

В. В. СЕМЕНЕЦ, канд. техн. наук, *В. Г. МАЛЮК*

АЛГОРИТМ ТРАССИРОВКИ ЦЕПЕЙ МНОГОСЛОЙНЫХ СТРУКТУР

При трассировке многослойных коммутационных схем на практике часто используются различные процедуры с послойной организацией поиска пути. Известно, что применение методов объемной трассировки существенно расширяет возможности повышения процента разведенных трасс. Область практического применения этих методов, однако, до настоящего времени была весьма узкой из-за значительных затрат времени и памяти при реализации их на ЭВМ.

Авторы существующих алгоритмов объемной трассировки многослойных соединений [1, 2] уделяли основное внимание проблеме памяти, остро стоявшей для ЭВМ старых поколений. Попытка преодолеть это ограничение приводила либо к большим затратам времени на обмен с внешней памятью ЭВМ [1], либо к значительному усложнению и повышению трудоемкости алгоритмов [2]. Развитие современных технических средств вычислительной техники в большинстве практических случаев позволяет снять жесткие ограничения на память. В связи с этим интересы разработчиков программного обеспечения этапов конструкторского проектирования сместились в сторону уменьшения трудоемкости алгоритмов трассировки и увеличения процента трассировки.

В данной статье описывается алгоритм построения соединений в многослойном дискретном поле (ДРП) в виде минимальных штейнеровских деревьев с помощью модификации волнового алгоритма, вычислительная сложность которого меньше, чем в известных процедурах объемной трассировки.

Пусть необходимо объединить некоторое множество контактов $M=\{m_1, ..., m_n\}$ в одну цепь с минимальной длиной. Для решения данной задачи применим алгоритм (в дальнейшем он будет именоваться алгоритмом 1):

Шаг 1. В качестве источника волны выбирается произвольный контакт $m_i \in M$.

Шаг 2. Волна распространяется до встречи с любым контактом $m_j \in M$, и между контактами m_i и m_j строится фрагмент цепи (m_i, m_j).

Шаг 3. Волна распространяется одновременно от всех ячеек $p \in (m_i, m_j)$ до встречи с очередным контактом данной цепи $m_\alpha \in M$.

Шаг 4. Проводится фрагмент цепи между контактом m_α и ранее проведенным фрагментом.

Последние два шага алгоритма повторяются до тех пор, пока все контакты данной цепи не будут соединены в цепь.

Введем понятие трудоемкости алгоритма. Так как в процедуре трассировки цепи около 80 % уходит на распространение волны, то критерием трудоемкости алгоритма может быть количество ячеек ДРП, которые являются источниками волны.

Алгоритм 1 имеет значительную трудоемкость уже при $n \geqslant 3$, так как после очередного пдосоединения контакта к цепи распространение волны осуществляется от всей цепи. Пусть за k этапов работы алгоритма построен фрагмент цепи длиной L_{k+1}. На $(k+1)$-м этапе к проведенному фрагменту будет добавлен фрагмент длиной L_{k+1}. Тогда трудоемкость $(k+1)$-го этапа выражается зависимостью

$$T_{k+1} = \int (L_k, L_{k+1}).$$

Полная трудоемкость алгоритма 1 будет равна

$$T_1 = \sum_{k=1}^{n} T_k.$$

Принцип работы алгоритма 1 показан на рис. 1. В верхней части ячейки ДРП указан номер фронта волны при распространении ее от контакта A, в нижней — от фрагмента AB.

Для уменьшения трудоемкости процесса построения минимальных связывающих деревьев разработан следующий метод (в дальнейшем он будет именоваться алгоритмом 2).

Шаг 1. В качестве источника волны выбирается произвольный контакт $m_i \in M$. Положить $k=1$.

		8/19	/19	/19	/18	/17	/16	/17	/18	/19				
	8/19	7/18	8/18	/18	/17	/16	/15	/16	/17	/18	/19			
8/19	7/18	6/17	7/17	8/17	/16	/15	/14	/15	/16	/17	/18	C/19		
7/18	6/17	5/16	6/16	7/16	8/15	/14	/13	/14	/15	/16	/17	/18		
6/17	5/16	4/15	5/15	6/15	7/14	8/13	/12	/13	/14	/15	/16	/17	/18	
5/16	4/15	3/14	4/14	5/14	6/13	7/12	8B	12	13	14	15	/16	/17	/18
4/15	3/14	2/13	3/13	4/13	5/13	6/12	7	/12	/13	/14	/15	/16	/17	/18
3/14	2/13	1/12	2/12	3/12	4/12	5/12	6	7/12	/13	/14	/15	/16	/17	/18
2/13	1/12	A	1	2	3	4	5	6/12	7/13	/14	/15	/16	/17	/18
3/14	2/13	1/12	2/12	3/12	4/12	5/12	6/12	7/13	/14	/15	/16	/17	/18	
4/15	3/14	2/13	3/13	4/13	5/13	6/13	7/13	/14	/15	/16	/17	/18		

Рис. 1. Принцип работы алгоритма 1

Шаг 2. Волна распространяется до тех пор, пока не достигнет всех контактов множества M или не заполнит полностью объем, охватывающий все контакты данной цепи. При этом, если волна достигает контакта $m_j \in M$, то в массив $D(1, k)$ записываются координаты x, y, z контакта, а массив $D(2, k)$ — номер фронта волны. Положить $k=k+1$.

Шаг 3. Проводится фрагмент цепи между контактами m_i и m_j, координаты которого хранятся в $D(1, 1)$. Положить $k=2$.

Шаг 4. Распространение вторичной объемной числовой волны по ячейкам ДРП, масса которых меньше, чем значение $D(2, k)$. Источником волны является контакт m_a с координатами x, y, z, взятыми из массива $D(1, k)$.

Шаг 5. При достижении ранее проведенного фрагмента строится путь минимальной длины между контактом m_a и данным фрагментом. Положить $k=k+1$. Если $k \leqslant n-1$, то идти к пункту 4, иначе КОНЕЦ.

В отличие от описанного выше алгоритма 1 процедуры распространения волны и проводки фрагмента цепи здесь разнесены.

Этап распространения вторичной волны является вспомогательным и существенно сокращает общую трудоемкость алгоритма.

На рис. 2 показан пример трассировки цепи, которой инцидентны три контакта — A, B, C. В качестве источника волны на первом этапе работы алгоритма выбран контакт A. Фрагмент (A, B) построен согласно шагу *3*. Контакт C является источником волны на втором этапе. В верхней части ячейки ДРП указан номер фронта волны при распространении ее от контакта A, в нижней — от контакта C.

Трудоемкость алгоритма 2 определяется выражением

$$T_{\text{II}} = T_0 + \sum_{k=2}^{n-1} T_k,$$

где T_0 — трудоемкость шагов 1—3: $T_0 \approx a \times b \times c$; a, b, c — размеры параллелепипеда, охватывающего все контакты данной цепи; T_k— трудоемкость k-го этапа. Очевидно, что трудоемкость $(m+1)$-го фронта волны равна $T_{m+1} = T_{m-1} + 4$, тогда

$$T_k = \begin{cases} (4p^2 + 3p + 1) \times c, & \text{если } L_k \text{ четное число;} \\ (4p^2 + 7p + 3) \times c, & \text{если } L_k \text{ нечетное число,} \end{cases}$$

где L_k — длина фрагмента цепи, который подсоединяется на k-м этапе к ранее проведенному фрагменту, $p = [L_k/2]$ — ближайшее целое.

Проиллюстрируем работу алгоритмов на следующих примерах:

Пример 1. Необходимо построить трассу цепи, которой инцидентны 20 контактов с координатами x и y соответственно: (45, 72), (61, 55), (22, 45), (18, 59), (7, 53), (11, 64), (13, 70), (8, 68), (2, 64), (36, 52), (45, 62), (50, 27), (37, 31), (24, 41), (11, 23), (13, 37), (35, 5), (53, 5), (55, 13), (59, 14). Размеры охватывающего параллелепипеда: $a = 63$, $b = 78$, $c = 10$. Результаты работы алгоритмов сведены в табл. 1. Время решения данного примера на ЭВМ ЕС 1050 алгоритмами 1 и 2 составило 62 с и 9 с соответственно.

Как видно из табл. 1, трудоемкость i-й итерации алгоритма 1 зависит от номера итерации. Так, на 1, 10, 16, 19 итерациях проводятся фрагмент длиной $L_1 = 18$, но $T_1 = 5530$, а $T_{19} = 46220$. Общая трудоемкость $T_{\text{I}} = 395560$, $L_{\text{I}} = 258$. Трудоемкость алгоритма 2 не зависит от номера итерации, а определяется только длиной фрагмента, который добавляется к ранее проведенной трассе: $T_{\text{II}} = 85240$, $L_{\text{II}} = 265$.

Пример 2. Необходимо построить цепь, которой инцидентны 10 контактов с координатами x и y соответственно: (3, 3), (8, 8), (6, 26), (14, 12), (14, 16), (20, 19), (27, 17), (32, 17), (26, 11), (29, 7). Определим трудоемкость каждого алгоритма при изменении начальной точки трассировки. Координаты запрещенных точек для трассировки: (6, 14), (29, 14), (16, 15)—(16, 25), (23, 15)—(23, 24), (5, 13)—(19, 13). Размеры обрамляющегося параллелепипеда: $a = 34$, $b = 29$, $c = 4$.

Результаты расчета сведены в табл. 2. Как видно из нее, трудоемкость каждого алгоритма практически не зависит от выбора

Таблица 1

Этап трассировки	Алгоритм 1: Трудоемкость i-го этапа в дискретах	Алгоритм 1: Номер контакта, включенного в цепь	Алгоритм 1: Длина фрагмента, подсоединяемого к цепи в дискретах	Алгоритм 2: Трудоемкость i-го этапа в дискретах	Алгоритм 2: Номер контакта, включенного в цепь	Алгоритм 2: Длина фрагмента, подсоединяемого к цепи в дискретах
0	—	1	—	48080	1	—
1	5530	2	18	—	2	18
2	5500	3	11	3240	16	19
3	6740	4	12	1960	8	14
4	4420	5	6	2100	3	16
5	3980	6	5	1980	14	18
6	7810	7	9	1320	4	12
7	11850	6	14	1820	13	13
8	14870	15	16	720	5	8
9	18200	14	15	1820	15	13
10	24010	8	18	180	6	4
11	28710	9	19	660	7	9
12	20690	10	10	3800	9	19
13	31880	11	16	3060	12	17
14	38730	13	21	2040	11	16
15	37400	12	17	1100	10	10
16	41620	19	18	6250	17	26
17	11650	20	4	3420	19	18
18	35750	18	11	180	20	4
19	46220	17	18	1510	18	11

Таблица 2

Начальная точка трассировки	Алгоритм 1: Трудоемкость в дискретах	Алгоритм 1: Длина трассы в дискретах	Алгоритм 2: Трудоемкость в дискретах	Алгоритм 2: Длина трассы в дискретах
1	42540	113	18040	123
2	43600	117	17120	125
3	42320	109	17930	123
4	44570	108	14570	106
5	44500	108	13840	106
6	45090	118	15160	107
7	43100	110	15540	114
8	42600	108	17120	123

начальной точки трассировки, однако алгоритм 2 в зависимости от выбора стартовой точки может построить трассу меньшей длины.

Следует заметить, что предлагаемый алгоритм объемной трассировки может быть использован как самостоятельно, так и в комбинации с процедурой плоской трассировки. В последнем варианте, сформировав объемное ДРП по результатам работы плоской трассировки, с помощью алгоритма 2 можно осуществить доразводку непроведенных соединений.

Алгоритм 2 реализован в виде программного модуля, предназначенного для работы в составе подсистемы конструкторского проектирования комплекса программ сквозного проектирования (КСПРЭА). Средства динамического резервирования памяти, используемые в комплексе, позволяют снять явные ограничения на длины рабочих массивов. Размерность проектируемой схемы лимитируется при этом лишь общим объемом памяти, распределяемой для решения задачи. В базовом варианте комплекса при объеме ОЗУ в 480 Кбайт с помощью данной программы объемной трассировки могут быть спроектированы до 4-х слоев платы размером 240×240 дискретов.

Список литературы: 1. *Heiss S.* A path connections algorithm for multilayer boards//Proc. of the 8-th Annual. 1969. P. 86—96. 2. *Geyer I. M.* Connection routing algorithm for printed circuit boards//IEEE Trans. 1971. CT-18. P. 95—100.

Поступила в редколлегию 21.07.86