

**В заключение** следует отметить, что рассмотренные в данной статье вопросы охватывают далеко не полный спектр возможностей, предоставляемых продукцией компании Cadence Design Systems. Технология Cadence охватывает практически все уровни разработки сложных систем – от системного уровня, свойственного разработчикам аппаратуры, до уровней логического, схемотехнического и топологического проектирования СБИС, их корпусирования, а также разработки печатных плат, на которых эти СБИС будут монтироваться.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Норенков И.П.* Основы автоматизированного проектирования / И.П. Норенков. – М.: Изд-во МГТУ имени Н.Э. Баумана, 2006. – 448 с.
2. *Курейчик В.М., Курейчик В.В., Родзин С.И.* Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-25.
3. *Стемпковский А.Л., Шепелев В.А., Власов А.В.* Системная среда САПР СБИС. – М.: Наука, 1994.
4. *Баринев С.В., Курейчик В.М.* Разработка алгоритма трехмерной компоновки СБИС на основе итерационной кластеризации // Проблемы разработки перспективных микро- и наноэлектронных систем – 2008. Сборник научных трудов / Под ред. А.Л. Стемпковского. – М.: ИПИМ РАН, 2008. – С. 55-61.
5. *Иванько А.В., Щеглов С.Н.* Основные этапы проектирования электронных устройств с использованием ПО компании CADENCE DESIGN SYSTEMS // Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'06) и «Интеллектуальные САПР» (CAD-2006). Научное издание в 3-х томах. – М.: Изд-во Физико-математической литературы, 2006. – Т. 2. – С. 49-56.
6. [www.cadence.com](http://www.cadence.com).
7. [www.crete.cadence.com](http://www.crete.cadence.com).

#### **Щеглов Сергей Николаевич**

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: [leo@tsure.ru](mailto:leo@tsure.ru).

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 8863371625.

Кафедра систем автоматизированного проектирования; к.т.н.; доцент.

#### **Shcheglov Sergej Nikolatvich**

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: [leo@tsure.ru](mailto:leo@tsure.ru).

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: +7863371625.

The Department of Computer Aided Design; Cand. of Eng. Sc.; Associate Professor.

УДК 681.31

**С.А. Ховансков, В.А. Литвиненко, О.Р. Норкин**

#### **ОРГАНИЗАЦИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ ТРАССИРОВКИ\***

*Предлагается подход к решению проблемы организации распределенных вычислений для выполнения объемных задач. В качестве вычислительной среды распределенных вычис-*

\* Работа выполнена при поддержке: РФФИ (грант № 10-01-90017), г/б № 2.1.2.1652.

*лений выбрано множество узлов компьютерной сети. Разработан алгоритм организации распределенных вычислений в этой среде для решения задачи трассировки. Определена конфигурация системы распределенных вычислений.*

*Распределенные вычисления; трассировка; многопроцессорная система; организация вычислений.*

**S.A. Hovanskov, V.A. Litvinenko, O.R. Norkin**

### **THE ORGANIZATION OF THE DISTRIBUTED CALCULATIONS FOR THE DECISION OF PROBLEMS OF TRACE**

*In article the approach to an organization solution of a problem raspredelennyh calculations for performance of volume problems is offered. As the computing environment of the distributed calculations the set of knots of a computer network is chosen. This environment the algorithm of the organization of the distributed calculations is developed for the decision of a problem of trace. The configuration of system of the distributed calculations is defined.*

*The distributed calculations; trace; mnogoprotses-weed system; the organization of calculations.*

В настоящее время использование и внедрение вычислительных систем (ВС) стало одним из показателей научно-технического прогресса. Постоянно разрабатываются новые, более совершенные ВС. Несмотря на возрастание производительности ВС, происходит более быстрый рост сложности задач требующих решения при соблюдении жестких временных ограничений. Поэтому разработка методов повышения быстродействия ВС и эффективности их использования при решении задач является актуальной задачей.

Одним из направлений повышения быстродействия ВС и соблюдения требуемых временных критериев решения задач является разработка методов организации распределенных вычислений, а направление развития и использования систем на основе автономных вычислительных узлов концентрирует все необходимые для таких вычислений возможности с наибольшей выразительностью и полнотой [1].

Среди наиболее трудоемких задач решаемых системами автоматизированного проектирования можно выделить задачи разработки новых узлов ВС, из которых основными по временным затратам являются задачи трассировки.

Для сокращения времени решения задач автоматизированного проектирования авторами статьи был разработан алгоритм трассировки, выполняемый путем организации распределенных вычислений на нескольких взаимодействующих вычислительных узлах [2].

В качестве вычислительных узлов для организации распределенных вычислений решения задач трассировки были выбраны ПК соединенные между собой в сеть. Выбор был произведен с точки зрения массовости и доступности узлов.

Известны различные методы решения задачи трассировки соединений, например, [4,5], предназначенные для трассировки на однопроцессорных вычислительных системах.

В работе [3] для трассировки соединений предложен алгоритм параллельного построения связывающих деревьев цепей на коммутационном рабочем поле. Этот алгоритм позволяет реализовать трассировку соединений с использованием в качестве вычислителя многопроцессорную систему. Его достоинствами являются простота, разрешение возникающих в ходе трассировки конфликтных и тупиковых ситуаций, а также учет расположения проведенных соединений. Алгоритм позволяет организовать одновременную работу всех процессоров, при этом каждый процессор работает независимо от остальных и трассировка всех соединений выполняется одновременно.

Алгоритм заключается в следующем. Все контакты цепи разбиваются на пары таким образом, чтобы при соединении этих пар контактов связями образовывалось дерево, включающее в себя все контакты.

Определим количество связываемых пар контактов  $S$  в множестве контактов  $M$ .

$$S = M - 1$$

Соответственно и число связей для построения дерева будет равно количеству пар контактов  $S$ .

Связи образуются путем присвоения последовательности дискретов метки цепи, начиная с исходного контакта. Из исходного контакта метка передается одному из четырех соседних дискретов – какому именно дискрету, определяется согласно последовательности приоритетных направлений. Этот дискрет включается в строящуюся цепь. Далее построение связи на дискретном рабочем поле между двумя дискретами, соответствующими начальному и конечному контактам производится путем аналогичной последовательной передачи и присвоением свободным дискретам метки цепи (рис. 1,2).

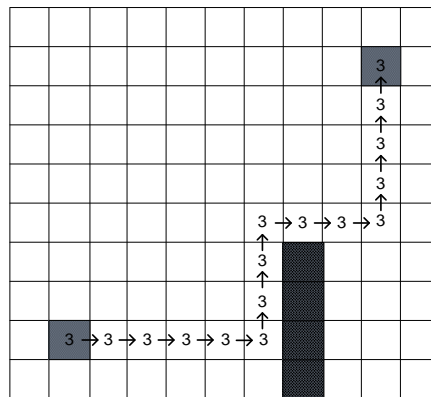


Рис. 1. Передача метки

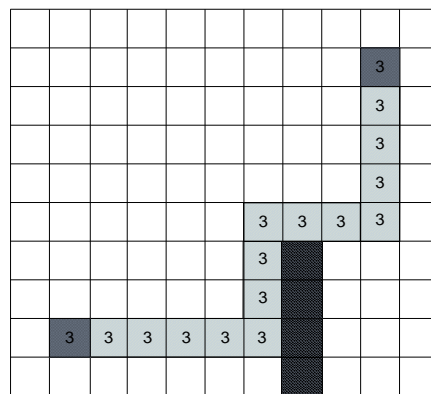


Рис. 2. Соединение контактов трассой

Алгоритм параллельного построения связывающих деревьев на коммутационном рабочем поле реализуется на МВС, имеющей решетчатую структуру и центральное устройство управления (ЦУУ). Множество узловых процессоров (УП)  $M_p$  образуют поле процессоров (ПП). Каждый процессор  $P(i, j) \in M_p$  может об-

мениваться информацией только с четырьмя соседними процессорами, определяемыми решетчатой структурой связей между процессорами.

Дерево строится на множестве дискретов  $D$  – рабочем поле коммутационного слоя. Рабочее поле  $D$  моделируется на решетчатом поле УП следующим образом. Каждому дискрету рабочего поля с координатами  $x, y$  ставится в соответствие УП  $P(i, j) \in M_p$ , где  $i, j$  – координаты УП в ПП. Каждый дискрет имеет координаты  $x, y$   $d(x, y) \in D$ . В общем случае, в каждый  $P(i, j) \in M_p$  может быть назначено  $\Delta x \times \Delta y$  дискретов рабочего поля  $D$ . Не нарушая общности, рассмотрим случай, когда  $\Delta x = \Delta y = 1$ . Обозначим  $M_{ij}$  множество процессоров, соседних процессору  $P(i, j)$ .

$$M_{ij} = \{P(i+1, j), P(i, j+1), P(i-1, j), P(i, j-1)\}.$$

Каждому дискрету  $d(x, y) \in D$  соответствует процессор  $P(i, j) \in M_p$ . Множество  $D_{xy}$  соседних дискретов для  $d(x, y)$  составит:

$$D_{xy} = \{d(x+1, y), d(x, y+1), d(x-1, y), d(x, y-1)\}.$$

Таким образом, между элементами  $M_{ij}$  и  $D_{xy}$  существует взаимнооднозначное соответствие.

Обозначим  $A$  – множество всех цепей  $A = \{A1, A2, \dots, Al, \dots, AL\}$ , где  $L$  – число цепей схемы. Для каждой цепи необходимо построить ортогональное связывающее дерево  $G_l$ . Контакты, которые входят в одну цепь  $Al$ , имеют метку  $l$ . Ребрами, образующими связывающее дерево, являются ортогональные связи, соединяющие пары контактов цепи.

Из пары соединяемых контактов один выбирается в качестве "источника", второй – в качестве "цели", соответствующие им УП помечаются аналогично. Из УП "источника" будет начинаться построение связи в соответствии с выбранными приоритетными направлениями. Задача состоит в построении связи из "источника" по направлению к "цели".

Построение связи заканчивается при достижении меткой, передаваемой через процессорное поле, узлового процессора с координатами "цели".

Для реализации алгоритма в каждый УП ЦУУ МВС засылает информацию о состоянии процессора и алгоритм его работы. Алгоритм, определяющий работу любого из процессоров, образующих процессорное поле, состоит из пяти этапов.

Перед началом построения во все  $P(i, j) \in M_p$  ЦУУ МВС засылает информацию о состоянии соответствующего ему дискрета  $d(x, y)$ , поскольку алгоритм построения определяет работу каждого УП в зависимости от его состояния. Состояния дискретов классифицируются следующим образом:

- ◆ свободный дискрет – дискрет, через который может пройти связь;
- ◆ запрещенный дискрет – дискрет запрещенный для построения через него связи;
- ◆ дискрет, которому соответствует контакт или часть связи цепи  $Al$ .

Обозначим  $S_{ij}$  – текущее состояние процессора  $P(i, j)$ .

В соответствии с состоянием дискрета состояние процессора  $P(i, j)$  кодируем следующим образом:

$$S_{ij} = \begin{cases} 0 - & \text{свободный УП;} \\ 1 - & \text{запрещенный;} \\ l - & \text{УП содержит часть связи цепи } A_l \text{ или контакт } k_{l\varphi}. \end{cases}$$

Кроме того, в каждый  $P(i, j) \in M_p [S_{ij}=1; k_{l\varphi} \in K^*]$  засылается информация о координатах "цели" –  $O_{l\varphi}^*$  соответствующих контакту.

Построение связи  $Z(l, m)$  заключается во включении на каждом шаге построения в последовательность процессоров, имеющих метку  $l$ , очередного процессора  $P(i, j)[S_{ij}=0]$  изменением его состояния на  $S_{ij}=l$ .

Для нахождения оптимального решения задачи построения дерева введем ограничение – деревья цепей должны иметь минимальное количество пересечений друг с другом. Это позволит более равномерно распределить связи по рабочему полю.

Связи строятся на поле УП МВС одновременно из всех источников за счет независимости работы каждого УП. Построение связи ведется путем передачи метки цепи из  $P_i$  соседнему процессору  $P_{i+1}$  и присвоение ему метки  $P_i$ .

Для минимизации общей длины связи будем производить построение дополнительных контактов в точках слияния связей дерева.

Процесс построения каждой связи  $Z(l, m)$  осуществляется до включения в последовательность процессоров, помеченных меткой  $l$  процессора  $P(i, j)$ , соответствующего контакту  $k_{l\varphi} \in K^*$  или  $P(i, j)[S_{ij}=l]$ , через который уже прошла связь  $Z(l, f) \in Z_l$ , причем, координаты цели связи  $Z(l, m)$  должны быть равны координатам источника связи  $Z(l, f)$ , а в месте слияния связей образуется новый контакт.

При решении задачи трассировки путем организации распределенных вычислений в компьютерной сети возникают трудности. Это связано с тем, что в отличие от МВС, полная автономность каждого вычислительного узла сети не позволяет при построении соединений приведенным алгоритмом использовать дискретное рабочее поле как единое целое.

Организации распределенных вычислений всегда сопровождается необходимостью обеспечения обменов между вычислительными узлами. В большинстве систем это реализуется с помощью выделенного вычислительного узла, управляющего работой всех остальных узлов системы. Либо путем использования механизма синхронизации работы всех вычислительных узлов. И тот и другой путь имеют существенный недостаток – наличие большого количества обменов, что, как правило, замедляет работу системы распределенных вычислений и может свести выигрыш по времени от использования распределенных вычислений к нулю.

Предлагается следующая реализация алгоритма параллельного построения связывающих деревьев цепей на коммутационном рабочем поле адаптированного в соответствии с учетом взаимодействия вычислительных узлов в сети и особенностями организации распределенных вычислений.

Конфигурация системы представляет собой граф компьютерной сети (рис. 3) с информационными потоками между узлами имеющий решетчатую структуру. Каждый узел обменивается информацией только с теми узлами, с которыми он связан.

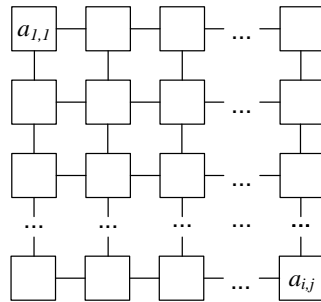


Рис. 3. Конфигурация системы

Поскольку процесс трассировки выполняется на нескольких вычислительных узлах, то перед началом процесса все множество дискретов  $D$  разбивается на соответствующее количество частей. Деление рабочего поля на части может выполняться либо с соблюдением критериев (например, размещение всех контактов трассы в одной части рабочего поля и др.), либо случайным образом. Они могут отличаться друг от друга размерами, геометрией.

На этапе разделения всего множества дискретов  $D$  между соседними частями определяются граничные линии, состоящие из множества дискретов  $d_i$  (рис. 4).

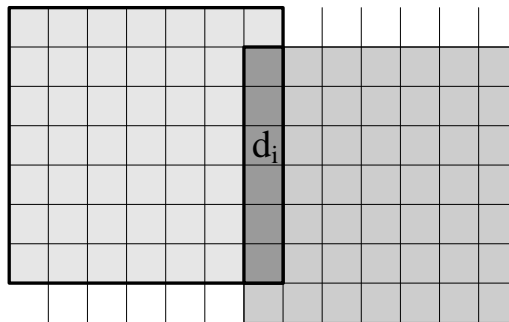


Рис. 4. Множество дискретов  $d_i$ , общих для двух узлов

Множество  $d_i$  является общим для двух соседних частей рабочего поля и оказывается принадлежащим соседним узлам.

В качестве исходной информации каждый вычислительный узел сети хранит все дискретное рабочее поле с выделением части определенной для данного узла и таблицу всех пар соединяемых контактов («источники» и «цели») с координатами их расположения на поле.

Процесс трассировки во всех узлах выполняется по приведенному выше алгоритму.

При выполнении построения связывающих деревьев цепей каждый вычислительный узел работает независимо от других, он последовательно связь за связью реализует процесс построения цепей в своей части дискретного рабочего поля.

Для снижения степени влияния автономности узлов сети на конечный результат алгоритм построения был изменен для учета фактора взаиморасположения соединяемых контактов.

Если контакты («источники» и «цели») расположены в одной части рабочего поля вычислительного узла  $m$ . Процесс построения связи с большой вероятностью будет реализовываться автономно одним вычислительным узлом. При этом обмен информацией с соседними узлами производиться не будет.

Другой случай, если контакты («источники» и «цели») расположены в разных частях рабочего поля. В этом случае выполняется построения связи до границы части рабочего поля вычислительного узла  $m$ .

Если в процессе построения связи, в последовательность дискрет помеченных меткой трассы включается дискрет  $g$  принадлежащий множеству  $di$ , то узел  $m$  путем передачи информации сообщает об этом соседнему узлу  $n$ , часть рабочего поля которого также включает множество  $di$ . После получения узлом  $n$  информации о том, что дискрет  $g$  принадлежащий множеству  $di$  помечен узлом  $m$ , изменяет свой список трассируемых связей. В список узла  $n$  добавляется новая нереализованная связь, «источник» которой располагается в дискрете  $g$ . Узел  $m$  считает трассу уже проложенной и занимается построением следующей по порядку связи.

Для продолжения этой трассы в соседней части рабочего поля процесс изменения списков трассируемых и протрассированных связей узлов повторяется. Благодаря организации обмена информацией любой узел, реализуя единый алгоритм построения связей, продолжит переходящее между частями соединение.

Предлагаемый алгоритм на распределенной вычислительной системе позволяет выполнить трассировку соединений любой длины. Благодаря одновременной реализации процесса построения связей несколькими вычислительными узлами, сокращается общее время трассировки. Для организации распределенных вычислений не требуется единого управляющего центра, количество обменов между узлами сведено к минимуму.

Достоинством такого подхода является то, что такая организация распределенной вычислительной системы позволяет реализовать выполнение трассировки на любом количестве узлов компьютерной сети. При этом нет определенных требований к рабочим узлам по производительности, по параметрам каналов связи. Необходимо только, чтобы объем памяти каждого из компьютеров позволял хранить всю исходную для трассировки информацию и программу трассировки. Количество узлов составляющих распределенную вычислительную систему не ограничено, но чем больше задействовано узлов в процессе трассировки, тем меньше время выполнения задачи.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ховансков С.А., Литвиненко В.А. Решения задач путем организации распределенных вычислений в сети // Известия ЮФУ. Технические науки. – 2008. – № 3 (80). – С. 16-21.
2. Ховансков С.А., Литвиненко В.А., Норкин О.Р. Оптимизации распределенных вычислений на базе алгоритма реконfigurирования и продукций предметной области // Труды Конгресса по интеллектуальным системам и информационным технологиям “AIS-IT’09”. – М.: Физматлит, 2009. – Т. 2. – С. 153-158.
3. Ховансков С.А., Литвиненко В.А., Калашиников В.А. Алгоритм трассировки на многопроцессорной вычислительной системе. – Депонировано в ВИНТИ 19.04.91 г., N1684-B91.
4. Курейчик В.М., Кажаров А.А. Муравьиные алгоритмы для решения транспортных задач // Известия РАН. Теория и системы управления. – М: Наука, 2010. – № 1. – С. 32-45.
5. Лебедев О.Б. Трассировка в канале методом муравьиной колонии // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 46-52.

#### **Ховансков Сергей Андреевич**

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: sah59@mail.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: +79185792173.

Кафедра радиоэлектронных средств защиты и сервиса; доцент.

**Литвиненко Василий Афанасьевич**

E-mail: litv@tsure.ru.

Тел.: 88634371651.

Кафедра систем автоматизированного проектирования; доцент.

**Норкин Олег Рауфатович**

Тел.: 88634371787.

Кафедра системного анализа и телекоммуникаций; доцент.

**Hovanskov Sergey Andreevich**

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: sah59@mail.ru.

Phone: +79185792173.

The Department of Radio-electronic Protection Frames and Service; Associate Professor.

**Litvinenko Vasily Afanasievich**

E-mail: litv@tsure.ru.

Phone: +78634371651.

The Department of Computer Aided Design; Associate Professor.

**Norkin Oleg Raufatovich**

Phone: +78634371787.

The Department of the System Analysis and Telecommunications; Associate Professor.

УДК 658.512

**И.Н. Ерошенко**

**МЕМЕТИЧЕСКИЙ АЛГОРИТМ ПЛАНИРОВАНИЯ СБИС\***

*Рассматривается задача планирования сверхбольших интегральных схем. Данная задача относится к классу NP. В работе предложен меметический алгоритм, являющийся гибридным генетическим алгоритмом. Приведено описание представления плана в виде обобщенной польской записи, к достоинствам которой относят быстрое декодирование и возможность негильотинного планирования. Также дано описание структур хромосом, принципы генетического и адаптивного поиска в рамках меметического алгоритма. Указана важность сбалансированного применения метаэвристик.*

*Планирование СБИС; меметический алгоритм; обобщенная польская запись.*

**I.N. Yeroshenko**

**MEMETIC ALGORITHM FOR VLSI FLOORPLANNING**

*The article is dedicated to the problem of VLSI floorplanning. VLSI floorplanning is an NP-hard problem. In this paper, a memetic algorithm as a hybrid genetic algorithm is presented. The description of floorplan in the form of generalized polish expression (GPE) is given. Fast decoding and possibility of nonslicing floorplan representation are the advantages of GPE. Chromosomes structures, genetic and adaptive search schemes within the memetic algorithm are proposed in this paper. The importance of the balanced usage of the metaheuristics is mentioned.*

*VLSI floorplanning; genetic algorithm; generalized polish expression.*

---

\* Работа выполнена при поддержке: РФФИ (грант № 09-07-00318), г/б № 2.1.2.1652.