Экспериментальный проект подсистемы конструкторского проектирования СБИС на основе иерархической клиент-серверной архитектуры

В.М. Глушань, О.И. Красюк, И.А. Дубровский

Инженерно-технологическая академия ЮФУ, г. Таганрог, gluval07@rambler.ru

Аннотация – Приводится описание разработки и испытаний опытного образца реальной подсистемы конструкторского проектирования СБИС, построенной на основе иерархической клиент-серверной архитектуры. Ее архитектура отличается от классической тем, что число компьютеров в ней равно числу компьютеров одного последнего уровня классической архитектуры. Работоспособность подсистемы проверялась при использовании реальных логических интегральных элементов при разном числе уровней иерархии. Приведен результат топологического проектирования СБИС подсистемой, состоящей из четырех компьютеров, а также временные зависимости от числа уровней подсистемы.

Ключевые слова – СБИС, иерархическая клиент-серверная архитектура, разбиение схемы, размещение, трассировка.

I. ВВЕДЕНИЕ

Задача топологического проектирования СБИС является одной из самых сложных среди всех задач, для решения которых привлекаются средства автоматизации. Это вызвано размерностью современных СБИС, измеряемой количеством транзисторов на кристалле полупроводника, число которых исчисляется уже миллиардами [1]. Для решения подобных сложных задач стали применять различные парадигмы распределенных компьютерных сетей, тем или иным образом объединяемых в кластеры [2]. В 70-80 гг. прошлого века активно проводились исследования по разработке спецпроцессоров (получивших название акселераторов) с целью ускорения решения задач конструкторского проектирования СБИС. Результаты некоторых из этих исследований отражены в монографии [3]. Но это направление отчасти по объективным, отчасти по социально-экономическим причинам постепенно угасло, а на повестку дня выдвинулись вычислительные сетевые технологии. С опорой на эти новые технологии в начале нынешнего столетия стали появляться разработки Web-ориентированных подсистем для географически распределенного конструкторского проектирования СБИС. Одна из таких разработок, построенная по принципу одноуровневой клиент-серверной архитектуры, была представлена в диссертации [4]. В этой подсистеме были реализованы только две проектные процедуры – декомпозиция (компоновка) и размещение элементов схемы в монтажном пространстве. Процедура трассировки межсоединений, являющаяся наи-

более трудоемкой при конструкторском проектировании, в этой подсистеме не была реализована. Это привело к ошибочному выводу, что использование большого количества компьютеров локальных или глобальных сетей может привести к выигрышу в быстродействии на порядок. Но это было верно, когда не решалась задача трассировки.

В последующих работах, обобщенно изложенных в монографии [5], было показано, что основанное на сетевых технологиях распределенное проектирование при включении в процесс конструкторского проектирования также и задачи трассировки дает максимальный эффект только при оптимальном числе компьютеров-клиентов, которое зависит от сложности схемы. Однако число их растет очень медленно даже при резком увеличении сложности схемы, и больше 3-4 компьютеров-клиентов использовать бессмысленно. При этом достичь выигрыша в быстродействии более, чем в 2-3 раза практически не удастся. Это заставило искать причину такого неэффективного использования компьютеров. Результатом этих поисков стала иерархическая клиент-серверная архитектура. В такой архитектуре увеличивалось – по сравнению с одноуровневой клиент-серверной архитектурой – число компьютеров-клиентов с ростом сложности проектируемой схемы. Это давало основание предполагать, что может быть достигнут и больший выигрыш в быстродействии. Эффективность иерархической-клиент серверной архитектуры изначально исследовалась на имитационных моделях [6-10]. Результаты имитационного моделирования подтвердили выдвинутую гипотезу и стали вдохновляющим стимулом для разработки опытного реального проекта подсистемы автоматизированного конструкторского проектирования СБИС. В данной статье приводится краткое описание этого проекта и результаты его возможностей для реального конструкторского проектирования.

II. ПРИНЦИП Р2Р-РАСПРЕДЕЛЕНИЯ ЗАДАЧ МЕЖДУ КОМПЬЮТЕРАМИ-КЛИЕНТАМИ

Как показано в [10], общее число компьютеров на всех уровнях в любой гомогенной иерархической архитектуре определяется по формуле для суммы геометрической прогрессии

$$S_n = 1 + \frac{q(q^n - 1)}{q - 1}, \quad (1)$$

где n – число уровней без учета нулевого уровня, q – цифра в обозначении уровней гомогенной архитектуры ($q = 2$ означает дихотомическую архитектуру, 3 – трихотомическую архитектуру и т.д.). Например, для шестилурневой архитектуры («222222») общее число компьютеров составит величину

$$S_6 = 1 + \frac{2(2^6 - 1)}{2 - 1} = 127.$$

Для гетерогенных архитектур расчет общего числа компьютеров осуществляется рекурсивно в соответствии с выражением

$$S_{\Sigma} = S_0 + S_1 + S_2 + S_3 + \dots = S_0 + \sum_{i=1}^{n_1} q_1^i + q_1^{n_1} \sum_{j=1}^{n_2} q_2^j + q_1^{n_1} q_2^{n_2} \sum_{k=1}^{n_3} q_3^k + \dots \quad (2)$$

В иерархической архитектуре на втором этапе работы подсистемы, т.е. после разбиения схемы на части, при работе компьютеров некоторого уровня простаивают компьютеры всех остальных уровней. Поэтому возникла задача устранения этого существенного недостатка. Решение этой задачи было найдено в равномерном распределении загрузки компьютеров по принципу «Peer-to-Peer», сокращенно – P2P, что означает «равный-к-равному» [11]. При этом сам принцип иерархичности в функционировании подсистемы не изменяется. Изменяется общее число компьютеров, и теперь оно определяется не по формуле (1), а как q^n , т.е. равно числу компьютеров последнего уровня обычной иерархической архитектуры. Поэтому такую модификацию архитектуры будем называть эквивалентной иерархической клиент-серверной архитектурой. Реализация такой архитектуры поясняется рис. 1.

Рис. 1. Пояснение загрузки компьютеров

На рис. 1 приведен пример для случая эквивалентной трехуровневой иерархической клиент-серверной архитектуры и общего числа узлов схемы, для удобст-

ва взятого равным 1024. Сразу отметим, что серым цветом отмечены компьютеры, получившие свою часть схемы в 128 элементов. Этот результат получен в процессе выполнения следующей процедуры (алгоритма). Исходно вся схема, состоящая из 1024 узлов, находится в главном сервере (компьютер 1). Половину этой схемы (512 узлов) сервер передает компьютеру 3. Передача частей схем на рис. 1 указана двунаправленными стрелками. Из оставшихся 512 узлов сервер опять отдает половину (256 узлов) компьютеру 2, а из оставшихся 256 узлов сервер отдает половину компьютеру 5. Таким образом, у главного сервера (компьютер 1) и у компьютера 5 оказалось по 128 узлов. Те компьютеры (2-й, 3-й и 4-й), у которых оказалось больше 128 узлов, аналогичным образом передают половину своей части схемы еще незагруженным компьютерам. Этот процесс продолжается до тех пор, пока все компьютеры не получают по 128 узлов. Если добавится еще один уровень из 8 компьютеров, то каждый из уже загруженных компьютеров передаст свою половину схемы одному из этих 8 компьютеров и т.д. Процесс сборки общей схемы будет происходить по направлениям обратных стрелок. Таким образом, все компьютеры будут работать одновременно до того момента, пока главный сервер не приступит к объединению частей схемы.

III. ОСНОВНЫЕ МОДУЛИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОДСИСТЕМЫ

Основными модулями подсистемы являются:

сетевой модуль – отвечает за поиск узлов (компьютеров) в локальной сети (по протоколу UDP) и за установку соединений и передачу данных (по TCP).

Модуль процедур (алгоритмов) – осуществляет генерацию схемы случайным образом, декомпозицию (разбиение), первичное и вторичное размещение, трассировку, композицию общей схемы (объединение).

При генерации схемы использовались реальные библиотечные логические элементы в интегральном исполнении, взятые из [12]. Пример логической принципиальной схемы и топологии одного из таких элементов приведен на рис. 2.

Рис. 2. Пример используемой логической схемы и ее топология

Исходными данными для генератора схем являются:

1. используемая библиотека логических элементов;

2. общее число элементов;
3. медиана (СКО) и пределы биномиального распределения функциональных узлов (числа элементов в них), заданные в степенях двойки. Так, медиана, равная 4, показывает, что в среднем функциональный узел содержит 2^4 радиоэлементов;
4. медиана и пределы биномиального распределения разветвленности схемы, которая показывает количество входных контактов, соединенных с соответствующим выходным;
5. вероятность p генерации «внутриузловой» связи. При генерации связи для какого-либо элемента функционального узла с вероятностью p будет сгенерирована связь с другим элементом этого же узла, и с вероятностью $1-p$ с любым другим узлом («межузловая» связь).

А. Декомпозиция

В качестве алгоритма декомпозиции использовался последовательный (конструктивный) алгоритм. Алгоритм получает на вход схему, в которой содержится список элементов и связей, и выдает список разбитых подсхем. Мы отказались от использования генетических алгоритмов по той причине, что они предполагают распараллеливание при работе с одним объектом. В нашем же случае подсистема работает параллельно со многими объектами.

В. Первичное размещение

Использовался последовательный алгоритм для назначения начальных позиций элементов схемы на сетке ДРП, поскольку четкие размеры позиций отсутствуют, а элементы могут иметь разные размеры. Входными данными алгоритма являются: схема, используемая для ее генерации библиотека (для определения размеров элементов) и коэффициент раздвижения, больший единицы. Поскольку используемые библиотечные элементы имеют одинаковую высоту и разную ширину, алгоритм производит размещение элементов с выравниванием по строкам. При этом если элементы имеют разную высоту, в начало строки будет помещаться элемент с наибольшей высотой.

С. Вторичное размещение

Для данной процедуры использовался итерационный алгоритм перестановок элементов. Он применялся для улучшения полученного на предыдущем этапе размещения. Поскольку элементы выровнены по строкам, алгоритм состоит из двух основных частей: групповой перестановки строк и парной перестановки элементов в пределах каждой строки.

Д. Трассировка

Для реализации данной процедуры использовался волновой алгоритм. В случае, если путь отсутствует, алгоритм производит раздвижение ДРП в точке волны, наиболее близкой к цели. Раздвижение может проводиться заданное количество раз. Если после исчерпания попыток связь не удастся провести, то она помечается как не проведенная, а затраченные попытки сбрасываются.

Е. Композиция

Алгоритм композиции используется для объединения всех частей схемы. При этом происходит объединение не только сеток, но и межсоединений между ними так, чтобы сделать их доступными последующей трассировке. Алгоритм также осуществляет попарные перестановки блоков ДРП для оптимизации их позиций. Целевая функция – сумма длин всех межсоединений, вычисляемая как разность координат контактов.

Для уменьшения пустых промежутков был добавлен поворот блоков на 90 градусов. Выполняется это так: для всех блоков берется отношение ширины к высоте. Если это отношение больше 1, блок разворачивается в ширину, меньше 1 – в высоту. В зависимости от того, какие блоки преобладают, остальные приводятся к такому же виду, с поворотом их на 90 градусов.

Модуль управления контролирует процесс проектирования, распределяя работу между узлами. Предусмотрены две основные схемы работы – стандартная иерархическая (частный случай – одноуровневый) и P2P. Организация такой схемы работы была описана выше. Не снижая производительность такой подход экономит используемые компьютеры, т.е. число компьютеров в многоуровневой иерархической архитектуре остается таким, каким оно было бы на последнем уровне стандартной иерархической архитектуры.

В сети используется два типа компонентов: менеджер и узел. Менеджер активен только на одном компьютере (сервер уровня 0), узлы активны на всех компьютерах сети (в том числе и на сервере уровня 0).

Интерфейс контроля работы подсистемы приведен на рис. 3 и 4.

IV. ОТЛАДКА И РЕЗУЛЬТАТЫ РАБОТЫ

Отладка системы осуществлялась на негетерогенных компьютерах с операционной системой Windows 7. После отладки программы были проведены исследования разработанной подсистемы. Использовались 8 гетерогенных компьютеров с процессорами Intel (R) Pentium CPU G3420 3,20 ГГц; ОЗУ 4,00 Гб; операционная система Windows 10, 64-разрядная. Результат работы подсистемы по генерации одной из четырех случайных схем соответствующим компьютером после решения задач размещения и трассировки приведены соответственно на рис. 5 и 6.

На трех других компьютерах одновременно проектировались три случайные подсхемы, которые затем были объединены сервером в одну общую схему. На рис. 7 приведен результат объединения сервером всех 4-х подсхем.

С целью сравнения реальных возможностей разных эквивалентных клиент-серверных архитектур подсистема запускалась на проектирование четырех схем с разным числом элементов и с разным числом уровней. Результаты этих экспериментов приведены на рис. 8.

Рис. 3. Интерфейс контроля работы клиентов

Рис. 4. Интерфейс контроля работы сервера

Анализ приведенных результатов позволяет сделать следующий вывод. При проектировании 4-х схем с разным числом элементов быстрее всех работала двухуровневая подсистема (2-2) – кривая 1 на графиках, и медленнее всех работала одноуровневая подсистема (8 компьютеров) – кривая 4. Причем двухуровневая система работала в 2,5 раза быстрее, чем одноуровневая подсистема. Это подтверждает теоретические результаты, полученные еще до начала работы над проектом. Промежуточное положение по скорости работы занимают одноуровневая (4 компьютера, график 3) и трехуровневая (2-2-2) подсистема (график 2). Экстраполируя эти две кривые, т.е. рассматривая случай, когда будут проектироваться схемы с большим числом элементов, можно утверждать, что кривая, полученная на трехуровневой подсистеме будет прохо-

дить ниже, чем на одноуровневой (при четырех компьютерах) подсистеме. Однако проведение экспериментов с большим числом элементов представляет трудности временного характера. Даже при числе элементов в схеме, равном 2000, одноуровневая подсистема (8 компьютеров) работала более 2-х часов.

V. ЗАКЛЮЧЕНИЕ

Проведенные исследования опытного образца подсистемы конструкторского проектирования СБИС, построенной по иерархической клиент-серверной архитектуре, позволяют достичь в несколько раз более высокого быстродействия по сравнению с одноуровневой подсистемой. Использование принципа P2P загрузки компьютеров оставляет их число таким, каким бы оно было в одноуровневой подсистеме, т.е. почти в

два раза меньшим. Как показывают подсчеты, выполненные по формуле (2), в случае шестиуровневой дихотомической архитектуры (т.е. 222222) вместо 127 компьютеров будет использоваться 64, а в случае шестиуровневой трихотомической архитектуры (т.е. 333333) вместо 729 компьютеров будет использоваться 1093. При проведении экспериментов было замечено, что основная доля времени приходится на объединение частей схем, когда сервер выполняет трассировку внешних связей. Это объясняется двумя причинами: во-первых, многие из внешних связей становятся более длинными по сравнению со связями, которые трассируют клиентские компьютеры (площадь квадрата или прямоугольника, с которой работает клиент, меньше). Во-вторых, число внешних связей при переходе от нижних уровней к верхним возрастает. Это также приводит к увеличению работы сервера. В это время клиентские компьютеры простаивают. Этот отрицательный эффект сводится к минимальной величине времени, когда проектируется схема, в которой число внутренних связей между элементами подсистемы оказывается меньшим числа всех внешних связей между подсистемами. Проведенные эксперименты выявили существование проблемных мест в разработанном опытным образом подсистеме как алгоритмического, так и программного характера. В то же время это позволило сформулировать пути устранения этих проблем, которые в перспективе могут привести к еще большему увеличению быстродействия таких подсистем.

Рис. 5. Одна из подсхем после размещения

Рис. 6. Подсхема после трассировки

Рис. 7. Объединенная схема

Рис. 8. Быстродействие подсистемы разных архитектур

ПОДДЕРЖКА

Работа выполнена при поддержке гранта РФФИ № 18-01-00041 «Разработка теоретических основ и принципов построения интеллектуальной распределенной подсистемы поддержки принятия оптимальных решений в задачах конструкторского проектирования на основе природных вычислений».

ЛИТЕРАТУРА

- [1] Тест и обзор: Intel Core i7-4770K и Core i5-4670K – новое поколение Haswell. URL: <http://www.hardwareluxx.ru/index.php/artikel/hardware/prozessoren/26018-haswell-test-intel-core-i7-4770k-core-i5.html> (дата обращения: 15.03.2018)
- [2] Воеводин Вл. В. Распределенная обработка данных. Вторая Сибирская школа-семинар по параллельным вычислениям/Под ред. проф. А.В.Старченко. – Томск: Изд-во Том. ун-та, 2004. – 108 с.
- [3] Курейчик В.М. и др. Комбинаторные аппаратные модели и алгоритмы в САПР/В.М. Курейчик, В.М. Глушань, Л.И. Щербаков. – М.: «Радио и связь», 1990. – 216 с.
- [4] Адамов А.З. Разработка и исследование распределенных WEB-ориентированных архитектур САПР. Дис. канд. техн. наук. – Таганрог, 2003. – 195 с.
- [5] Глушань В. М., Лаврик П. В. Распределенные САПР. Архитектура и возможности. Монография. Старый Оскол: ТНТ, 2014. – 188
- [6] Глушань В.М., Лаврик П.В. Возможности распределенной подсистемы топологического проектирования, построенной на основе клиент-серверных технологий. Проблемы разработки перспективных микро-и нано электронных систем. – 2014. Сборник трудов/Под общей ред. Академика РАН А.Л. Стемповского. М.: ИППМ РАН, 2014, Ч. II. С. 11-14.
- [7] Glushan V.M., Lavrik P.V., Rybalchenko M.V. Hypergraph model of hierarchical client-server architecture for distributed computing. 9 th International Conference Application of Information and communication technologies – AICT2015. October 2015, Rostov-on-Don, pp.454-457.
- [8] Глушаь В.М., Лаврик П.В., Рыбальченко М.В. Исследования модели распределенного топологического проектирования СБИС средствами иерархической клиент-серверной архитектуры // Проблемы разработки перспективных микро- и наноэлектронных систем - 2016. Сборник трудов / под общ. ред. академика РАН А.Л. Стемповского. М.: ИППМ РАН, 2016. Часть II. С. 190-196.
- [9] Glushan V.M., Lavrik P.V., Lozovoy A.Yu., Rybalchenko M.V. Distributed mode of topological design of VLSI by means of hierarchical client-server architecture. Atlantis Press, 2016. Amsterdam-Hong-Kong-Paris. Information Technologies in Science, Management, social Sphere and Medicine (ITSMSM 2016), pp. 532-536.
- [10] В.М. Глушань, И.А. Дубровский, О.И. Красюк, М.В. Рыбальченко. Анализ трудоемкости конструкторского проектирования СБИС средствами гибридной иерархической клиент-серверной подсистемы/Вестник РГРТУ, 2017, № 62, с. 71-78.
- [11] Радченко Г.И. Распределенные вычислительные системы / Г.И. Радченко. – Челябинск: Фотохудожник, 2012. – 184 с.
- [12] URL:http://www.vlsitechnoogy.org/html/cells/vsclib013/lib_gif_index.htm (дата обращения: 23.03.2018)

Experimental Project of the of VLSI Design Subsystem on the Basis of Hierarchical Client-Server Architecture

V.M. Glushan, O.I. Krasnyuk, I.A. Dubrovsky

Engineering and Technological Academy, Southern Federal University

gluval07@rambler.ru

Abstract — The early emulation studies of the VLSI design subsystem constructed on the basis of the hierarchical client-server architecture confirmed the hypothesis of the possibility of obtaining higher performance compared to the single-layer architecture. The article presents the results of the construction and investigation of the real prototype of the VLSI design subsystem. To study the performance of the subsystem, basic circuits were generated, containing from 500 to 2.000 functional logic elements with a random specification of the circuit parameters: the number of elements, the number of contacts per the element, and the circuit branching. During the research of the subsystem, real integral logic elements were used. Each circuit was designed by a subsystem with a different number of levels: from one to four, and the time of the subsystem operation was compared. As a result, it was discovered that the initially planned study of the heterogeneous subsystem in the first experiments has given negative result. Therefore, in all following studies only homogeneous subsystems were used. Studies have shown that depending on a hierarchical client-server architecture, you can obtain performance several times better than with the one-level ones. And using the computers based on the principle of "Peer-to-Peer" leads to almost twofold decrease in the number of computers compared to the standard client-server architecture.

Keywords — homogeneous, hierarchical client-server architecture, VLSI, schema partitioning, placement, routing.

REFERENCE

- [1] Test i obzor: intel core i7-4770k i core i5-4670k – no-voepokolenie haswell. url: <http://www.hardwareluxx.ru/index.php/artikel/hardware/prozessoren/26018-haswell-test-intel-core-i7-4770k-core-i5.html> (data obrashcheniya:15.03.2018)
- [2] Voevodin V.I. V. Raspredeennaya obrabotka dannyh. Vtoraya Sibirskaya shkola-seminar po parallelnym vychisleniyam/Pod red. prof. A.V.Starchenko. – Tomsk: Izd-vo Tom. un-ta, 2004. – 108 S.
- [3] Kurejchik V.M. i dr. Kombinatornye apparatnye modeli i algoritmy v SAPR/V.M. Kurejchik, V.M. Glushan', L.I. Shcherbakov. – M.: «Radio i svyaz'», 1990. – 216.
- [4] Adamov A.Z. Razrabotka i issledovanie raspredelen-nyh WEB-orientirovannyh arhitektur SAPR. Dis. kand. tekhn. nauk. – Taganrog, 2003. – 195 s.
- [5] Glushan' V. M., Lavrik P. V. Raspredelennye SAPR. Arhitektura i vozmozhnosti. Monografiya. Staryj Oskol: TNT, 2014. – 188.
- [6] Glushan' V.M., Lavrik P.V. Vozmozhnosti raspredelennoy podsystemy topologicheskogo proektirovaniya, postroennoy na osnove klient-servernyh tekhnologij. Problemy razrabotki perspektivnyh mikro- i nano ehlektronnyh sistem. – 2014. Sbornik trudov/Pod obshchey red. Akademika RAN A.L. Stempkovskogo. M.: IPPM RAN, 2014, CH. II. S. 11-14.
- [7] Glushan V.M., Lavrik P.V., Rybalchenko M.V. Hypergraph model of hierarchical client-server architecture for distributed computing. 9 th International Conference Application of Information and communication technologies – AICT2015. October 2015, Rostov-on-Don, pp.454-457.
- [8] Glushan' V.M., Lavrik P.V., Rybal'chenko M.V. Issledovaniya modeli raspredelennoy topologicheskogo proektirovaniya SBIS sredstvami ierarhicheskoy klient-servernoy arhitektury // Problemy razrabotki perspektivnyh mikro- i nanoehlektronnyh sistem - 2016. Sbornik trudov / pod obshch. red. akademi-ka RAN A.L. Stempkovskogo. M.: IPPM RAN, 2016. CHast' II. S. 190-196.
- [9] Glushan V.M., Lavrik P.V., Lozovoy A.Yu., Rybalchenko M.V. Distributed mode of topological design of VLSI by means of hierarchical client-server architecture. Atlantis Press, 2016. Amsterdam-Hong-Kong-Paris. Information Technologies in Science, Management, social Sphere and Medicine (ITSMSM 2016), pp. 532-536.
- [10] Glushan' V.M., Lavrik P.V., Rybal'chenko M.V. Issledovaniya modeli raspredelennoy topologicheskogo proektirovaniya SBIS sredstvami ierarhicheskoy klient-servernoy arhitektury // Problemy razrabotki perspektivnyh mikro- i nanoehlektronnyh sistem - 2016. Sbornik trudov / pod obshch. red. akademi-ka RAN A.L. Stempkovskogo. M.: IPPM RAN, 2016. CHast' II. S. 190-196.
- [11] Radchenko G.I. Raspredelennye vychislitel'nye si-stemy / G.I. Radchenko. – Chelyabinsk: Fotohudozhnik, 2012. – 184 s.
- [12] URL:http://www.vlsitechnoogy.org/html/cells/vsclib013/lib_gif_index.htm (access date 23.03.2018)