

УДК 519.7, 519.17

ИСПОЛЬЗОВАНИЕ ГРИД-СИСТЕМ ДЛЯ ВЫПОЛНЕНИЯ ТРУДОЕМКИХ ОПЕРАЦИЙ ПРОЕКТИРОВАНИЯ СБИС



Д.И. Черемисинов

Ведущий научный сотрудник ОИПИ НАНБ
кандидат технических наук, доцент
cher@newman.bas-net.by



Л.Д. Черемисинова

Главный научный сотрудник ОИПИ НАНБ
доктор технических наук, профессор
cld@newman.bas-net.by

Объединений институт проблем информатики Национальной академии наук Беларуси, Республика Беларусь.
E-mail: {cher, cld}@newman.bas-net.by.

Д. И. Черемисинов

Окончил Томский государственный университет, кандидат технических наук, доцент. Работает в ОИПИ НАН Беларуси в должности ведущего научного сотрудника и Белорусском государственном университете информатики и радиоэлектроники в должности доцента.

Круг научных интересов: программирование, логическое проектирование и тестирование дискретных систем управления, реализация параллельных алгоритмов управления.

Л. Д. Черемисинова

Окончила Томский государственный университет, доктор технических наук, профессор. Работает в ОИПИ НАН Беларуси в должности главного научного сотрудника и Белорусском государственном университете информатики и радиоэлектроники в должности профессора. Круг научных интересов: дискретная математика, логическое проектирование и тестирование дискретных систем управления, реализация параллельных алгоритмов управления.

Аннотация. Рассматривается проблема ускорения трудоемких расчетов при решении комбинаторных задач проектирования СБИС путем выполнения распределенных вычислений на удаленных суперкомпьютерах. Описывается методика использования сервиса грид-системы для выполнения трудоемких расчетов, основанная на применении открытых стандартов и межплатформенного связующего программного обеспечения, являющегося посредником между программами, выполняемыми на удаленных компьютерах. Предлагаются программные средства, обеспечивающие совместимость по данным и организацию обмена данными между грид-системой и клиентской системой автоматизации проектирования, позволяющие избежать утечки информации о назначении и характеристиках проектируемого устройства.

Ключевые слова: грид-система, распределенные вычисления, автоматизация проектирования СБИС, конверторы форматов описаний.

Введение.

Современные сверхбольшие интегральные схемы (СБИС) могут содержать более сотни млн. транзисторов, и примерно такое же количество соединений между ними. Процесс подготовки производства современной СБИС стоит очень дорого: только изготовление набора фотошаблонов требует затрат в несколько миллионов долларов. С увеличением сложности СБИС растет сложность задач их проектирования и верификации. Проектирование цифровых систем на микроэлектронной базе СБИС представляет собой сложнейшую научно-техническую проблему в связи с постоянно уменьшающимися проектными технологическими нормами и увеличивающимися требованиями к их надежности функционалу.

Проектирование современных СБИС в силу его сложности возможно только с использованием современных систем автоматизации проектирования (САПР). Решаемые САПР задачи проектирования СБИС уже зачастую требуют десятков часов процессорного времени современных

компьютеров, что является неприемлемым для интерактивной отладки проекта. Алгоритмы САПР можно ускорить с помощью высокопроизводительных вычислительных систем общего назначения, доступных через суперкомпьютерные центры, которые предоставляют доступ к суперкомпьютерам широкому кругу пользователей, подключая их машины к сети и разрешая удаленный доступ через Интернет посредством службы WWW. Такие веб-центры САПР СБИС могут дать доступ к инструментам САПР для проектирования СБИС дизайнерским центрам, а также позволить разработчикам, рассредоточенным географически, использовать единый набор инструментов и совместно работать над проектом сложной СБИС. Совместное проектирование становится все более важной задачей, потому что интеграция систем на одном чипе требует большого разнообразия навыков и знаний, которые не всегда можно найти в одной организации.

Процесс проектирования СБИС состоит из этапов, в процессе выполнения которых исходное задание на изготовление СБИС в виде алгоритмических спецификаций на высокоуровневых языках проектирования типа VHDL и Verilog [1] преобразуется в топологические описания, по которым изготавливаются кристаллы. Наиболее наукоемким этапом проектирования являются алгоритмическое и логическое проектирование в силу того, что решаемые на этом этапе задачи принадлежат к классу труднорешаемых комбинаторно-логических. В то же время от решения этих задач существенно зависят основные характеристики получаемых кристаллов, такие как площадь, быстродействие, энергопотребление и надежность.

Потребность в решении комбинаторных задач возникает во многих областях, в том числе таких, как автоматизированное проектирование, создание систем искусственного интеллекта, разработка сетей связи и криптография. Под комбинаторно-логическими задачами подразумеваются перечислительные и поисковые задачи на конечных множествах, элементами которых служат объекты, представляющие собой комбинации элементов других конечных множеств – сочетания, перестановки, разбиения, покрытия, решения систем логических уравнений и т. п. Особенностью задач рассматриваемого класса является то, что объемы перерабатываемой информации при поиске решений относительно невелики, а сами процессы переработки достаточно сложны. Почти все комбинаторно-логические алгоритмы имеют экспоненциальную сложность. Ввиду увеличивающейся сложности интегральных схем новых поколений, непрерывно возрастают требования к инструментам автоматизированного проектирования. Большинство задач проектирования интегральных схем для параметров сложности, встречающихся в практике, находятся на грани возможностей современной вычислительной техники. Один из вариантов решения данной проблемы лежит в распараллеливании алгоритмов проектирования и выполнения расчетов с помощью сервисов Грид-системы.

Веб-центр САПР СБИС.

Программное обеспечение веб-центра САПР СБИС имеет архитектуру веб-приложения. Веб-приложения можно разделить на несколько типов, в зависимости от разных сочетаний его основных составляющих: Backend (серверная часть приложения) работает на удаленном компьютере, который может находиться где угодно и выполняет сложные операции проектирования; Frontend (часть приложения) выполняется в браузере пользователя, где и собирается проект СБИС в целом.

Современные грид-системы строятся на базе программного обеспечения, предназначенного для объединения распределенных ресурсов в рамках единой виртуальной организации. Важной проблемой, препятствующей реальному использованию веб-центра САПР СБИС, является безопасность данных, которые представляют значительную ценность интеллектуальной собственности. Проблемы безопасности веб-центра САПР СБИС можно разбить на три категории: безопасность клиента, безопасность передачи и безопасность сервера.

Безопасность клиента охватывает все аспекты защиты данных на компьютере, не связанные с веб-системой, и при выполнении алгоритма САПР на стороне клиента отсутствует угроза интеллектуальной собственности. Защита передачи данных между веб-сервером и клиентской частью веб-приложения может осуществляться с помощью шифрования с открытым ключом, такого как в протоколе SSH (Secure Shell, «безопасная оболочка») [2]. Все сообщения от пользователя к вычислительному центру будут зашифрованы открытым ключом центра, а обратная связь использует открытый ключ пользователя. Таким образом, проблемы безопасности клиента и безопасности

передачи решаются с помощью стандартных решений Интернет-систем.

Использование веб-приложений требует доверия к серверу, и проблемы безопасности данных на сервере не зависят от протокола коммуникации. Если сервер находится под контролем противника, опасность потери интеллектуальной собственности неустранима, и необходимо предпринять шаги, чтобы минимизировать объем информации, проходящей между веб-центром САПР СБИС и рабочей станцией проектировщика. Однако, выполнение в веб-центре таких этапов проектирования как логический синтез и логическое моделирование, требуют существенной информации о проектируемой схеме, для скрытия которой есть только ограниченные возможности. Предложенные методы защиты проекта СБИС в веб-САПР не способны обеспечить безопасность [3]. Например, в САПР Cadence для этапа логического моделирования предложен метод защиты шифрованием описания моделируемой схемы. Однако эта методология защищает исходные файлы от прямого наблюдения (гарантируя безопасность передачи), но не защищает информацию в самих программах САПР Cadence.

Основная проблема безопасности веб-САПР заключается в том, что пользователю приходится передавать информацию на суперкомпьютер, чтобы выполнить оптимизацию схемы, но нужны гарантии, что при этом другим невозможно получить представление о назначении и характеристиках проектируемого устройства. Решение этой дилеммы состоит в том, чтобы предоставлять в веб-центр только необходимую для оптимизации проектных решений информацию. Функциональность устройства может быть скрыта, если в веб-центре выполняются не операции проектирования, а решаются оптимизационные задачи, лежащие в их основе.

Веб-САПР для проектирования интегральных схем.

Задачи логической оптимизации структуры интегральных схем носят комбинаторный характер и имеют экспоненциальную сложность, именно поэтому они и нуждаются в выполнении быстрых распределенных вычислений. В то же время программы логической оптимизации в качестве ресурса Веб-САПР позволяют обеспечить сохранность интеллектуальной собственности проекта интегральной схемы, поскольку имеют в основе постановку и решение абстрактной задачи. Часто оптимизация состоит в решении задач, сводящихся к удовлетворению ограничений, в частности булевых ограничений, представляемых в виде систем булевых уравнений.

В основе архитектуры некоторых широко используемых САПР цифровых схем (таких как системы LeonardoSpectrum синтеза схем и ModelSim моделирования [4]) лежит модульный принцип организации выполнения проектных операций. Порядок выполнения и обмен данными между модулями задается на языке управления TCL (Tool Command Language) и есть возможность извне управлять процессами проектирования путем написания скриптов на языке TCL. Такая архитектура САПР дает возможность разрывать цикл проектирования в рамках САПР, решать наиболее сложные оптимизационные задачи проектирования и верификации вне САПР и интегрировать затем полученные решения в цикл проектирования.

Действия по передаче данных, связанных с решением абстрактной оптимизационной задачи, можно выполнить посредством межплатформенного связующего программного обеспечения (middleware) грид-сетей, занимающее «промежуточное» положение между приложениями пользователей и непосредственно суперкомпьютером. В национальной грид-сети Республики Беларусь связующим слоем – middleware – является интерфейс для распределенных вычислений UNICORE [5, 6], стандартный интерфейс доступа клиентских частей грид-сервиса к вычислительным ресурсам на суперкомпьютере. Пользовательский уровень и UNICORE сервер связаны открытой сетью и взаимодействуют между собой через протокол SSL (Secure Socket Layer, уровень защищённых сокетов).

Задача компонента пользовательского уровня UNICORE состоит в подготовке UNICORE задания для суперкомпьютера и отслеживании процесса выполнения суперкомпьютерной программы. Через интерфейс пользовательского уровня UNICORE пользователь указывает параметры задачи: суперкомпьютер, на котором должно выполняться задание, командную строку запуска суперкомпьютерной программы, файлы для импорта и экспорта. Трансляция абстрактного определения задачи в формат пакетного задания для реального суперкомпьютера также происходит

через сетевой диспетчер UNICORE (рисунок 1).

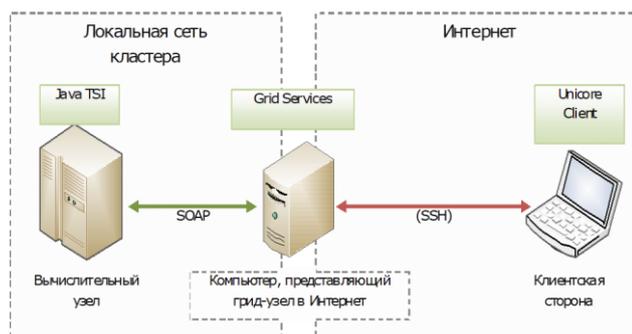


Рисунок 1. Взаимодействие компонент UNICORE

Кластерный суперкомпьютер в общей модели UNICORE представлен как виртуальный сайт с единым файловым пространством. Каждый сайт состоит из интерфейса к локальной операционной системе и пакетной системы управления заданиями. UNICORE использует стандарты, основанные на XML во всех функциональных областях: протокол SOAP (Simple Object Access Protocol, простой протокол доступа к объектам) для обмена структурированными сообщениями в распределённой вычислительной среде для связи пользовательского уровня и сервера; язык JSDL (Job Submission Description Language, язык представления заданий) для представления работы; язык SAML (Security Assertion Markup Language, язык задания декларации безопасности) для аутентификации и авторизации между пользовательским уровнем и сервером; язык XACML (eXtensible Access Control Markup Language, язык задания контроля доступа) для выполнения аутентификации и авторизации между сервером и суперкомпьютером. Средства пользовательского уровня UNICORE позволяют осуществлять взаимодействие пользователя со всей грид-системой: ставить в очередь задачи, получать информацию об их статусе, передавать файлы между локальным и удалённым компьютерами.

Трансформация форм описаний структуры интегральной схемы.

Функционально-структурное описание проектируемой СБИС на разных этапах проектирования является иерархическим. В иерархических описаниях схем на языках спецификации оборудования, общие подсхемы группируются вместе, с тем, чтобы облегчить понимание функций и назначения частей устройства и сжать описание проектируемой схемы в целом. Но для выполнения оптимизационных преобразований используется плоское описание оптимизируемого блока, в котором отсутствует иерархия. Например, при логическом синтезе одними из основных оптимизационных задач являются минимизация и декомпозиция систем булевых функций. В этом случае перед решением задач оптимизации выполняется удаление уровней иерархии путем замены вызовов подсхем в самой логикой. В полученном плоском описании схемы без содержательных названий цепей информация о функциях и назначении устройства в целом содержится только частично.

Для применения такого метода скрытия информации о функциональности и назначении проектируемого устройства в информации, передаваемой в веб-центр, необходимо использовать программы преобразования форматов данных, используемых в САПР СБИС, до того, как информация будет отправлена в веб-центра, а также для обратного преобразования результатов работы оптимизационных программ веб-центра.

Местом развертывания конверторов может быть и клиентская, и серверная сторона веб-САПР. С точки зрения безопасности предпочтительнее развёртывание конверторов на клиентской стороне.

Конверторы представлений данных.

Проблема конвертации описания на формальном языке имеет большое практическое значение в области технологии программирования, называемой «реинжинирингом программного обеспечения» (software reengineering). Одной из наиболее распространенных форм реинжиниринга

являются языковые преобразования, подразумевающие преобразование текста программ в эквивалентные им по функциональности тексты на целевом языке высокого уровня. Соответствующая операция преобразования может выполняться прямо в ходе анализа исходного текста. Операция интерпретации является гомоморфизмом, т. е. между структурами исходного и целевого описания выполняется следующее соотношение:

$$f(\text{op}(a, b)) = \text{op}*(f(a), f(b)).$$

Интерпретация f структуры op (из исходного описания) с компонентами a и b определяется как новая структура op^* (целевого языка), компоненты которой определяются применением интерпретации f к a и b . В программировании структуры, упомянутые выше, представлены типами данных. В гомоморфной интерпретации разрешается выполнять не инъективное отображение, то есть, различные структуры исходного описания могут быть отображены на одинаковые структуры в целевом описании, что обозначает отсутствие однозначного способа преобразования типа op в тип op^* .

Программы разработанных конверторов форматов данных построены так, чтобы их код не требовал глобального изменения при замене целевого языка. Эта цель достигается разбиением интерпретации f на две части: операцию анализа структур исходного языка с фиксацией результата в промежуточной структуре и операцию обхода промежуточной структуры с генерацией представления на целевом языке (рисунок 2).

Разбиение интерпретации приводит к структуре программы, состоящей из компоненты импорта, выполняющей анализ, и компоненты экспорта, выполняющей сериализацию (сохранение объектов в долговременной памяти в период выполнения программы) промежуточной структуры в целевое описание. Компоненты импорта и экспорта взаимодействуют через интерфейс, представляющий собой промежуточный объект HUB (ядро). В такой архитектуре смена исходного или целевого языка ведет к изменению только одной компоненты: импорта или экспорта (рисунок 2.).

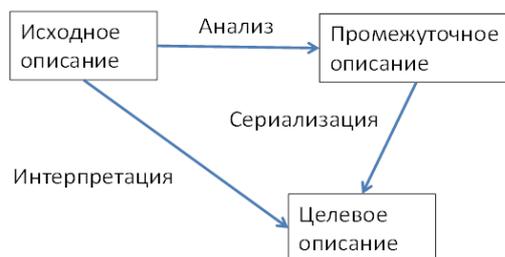


Рисунок 2. Разбиение интерпретации на две операции

Конвертор описаний структуры цифрового устройства, как и компилятор, имеет три компонента: 1) front-end, который анализирует входной текст, 2) собственно транслятор, который выполняет преобразование, не зависящее от формы исходного и целевого представлений, и 3) back-end, который генерирует текст результата компиляции. Компонент front-end конвертера реализует программу импорта описаний структуры цифрового устройства.

Анализатора языка front-end традиционно включают этап лексического и синтаксического анализа [7]. Его задачей является построение структуры анализируемого текста путем выполнения его разбора. Алгоритм распознавания принадлежности текста заданному языку имеет форму грамматики языка. Работа синтаксического анализатора состоит в построении структуры HUB, соответствующей исходному описанию. Компонент экспорта back-end представляет собой объект, называемый форматером, который и выполняет нужную сериализацию.

Предлагаемый набор конверторов форматов описаний.

Наиболее часто используемыми в САПР интегральных схем являются формат EDIF [8] и структурный и поведенческий язык VHDL [9]. Чтобы интегрировать в сквозной процесс

проектирования программы, выполняющиеся в Веб-САПР, достаточно иметь средства преобразования между тремя форматами: EDIF, VHDL и языком представления систем булевых функций. В качестве последнего предлагается использовать языки SF и BLIF. Язык SF является внутренним языком программных средств логического проектирования дискретных устройств управления, разрабатываемых в ОИПИ НАН Беларуси [10]. Этот язык позволяет описывать функционально-структурные описания дискретных устройств, в нем имеются форматы задания систем булевых функций в матричном виде и в виде логических уравнений. Формат BLIF (Berkeley Logic Interchange Format) используется для задания систем булевых функций в открытых программных средствах оптимизации и синтеза логических схем SIS и ABC [11].

Формат EDIF принят в качестве стандарта для обмена информацией в САПР электронных устройств. В настоящее время наиболее широко используется версия 2.0.0 этого формата, позволяющая описывать структуру проектируемых устройств. Анализатор EDIF версии 2.0.0, реализованный в конвертере описаний на EDIF в описания на языке SF (и обратно), построен трансляцией грамматики EDIF, свободно доступной через Веб-сервер Манчестерского университета в Англии [8].

Язык VHDL (Very high speed integrated circuits Hardware Description Language) является формальным языком описания логических схем для всех этапов разработки аппаратуры современных вычислительных систем. Первоначально язык предназначался для моделирования, но позднее из него было выделено синтезируемое подмножество. Анализатор VHDL, реализованный в конвертере, построен трансляцией грамматики языка VHDL «Стандарт VHDL-93» [9]. Все конструкции языка допустимы, но конвертером обрабатывается только синтезируемое подмножество языка, то есть не обрабатываемые конструкции игнорируются. Это делает описания, представленные на этом подмножестве, пригодными для использования с любым коммерческим или бесплатным инструментом VHDL. Обрабатываемое подмножество позволяет описывать только цифровые синхронные схемы. Исходное описание должно соответствовать структурному стилю, допускается включение операторов сигнального присваивания с тривиальной правой частью. Обрабатываемые описания должны принадлежать синтезируемому подмножеству VHDL по типам сигналов. При преобразовании из других форматов могут строиться описания VHDL поведенческого стиля.

Формат SF является языком иерархического функционально-структурного описания логических схем и особенно удобен для описания и оптимизации комбинационной логики. Конвертер обрабатывает текстовые описания формата SF, содержащие и структурные, и функциональные описания схем, в том числе файлы, содержащие несколько описаний различных типов.

Формат BLIF (Berkeley Logic Interchange Format) [11] предназначен для описания иерархических схем на логическом уровне в текстовой форме. Здесь схема представляет собой произвольную комбинационную или последовательную сеть логических функций, которая может рассматриваться как ориентированный граф комбинационных логических и последовательных элементов.

Разработаны конверторы, реализующие попарные преобразования перечисленных выше форматов [12].

Заключение.

В настоящей работе предлагается архитектура сервиса грид-системы для выполнения трудоемких расчетов при проектировании СБИС. Сервисный подход для управления работой программы удаленного компьютера привлекателен тем, что ориентирован на использование открытых, универсальных протоколов и интерфейсов. Предлагается методика и программные средства, которые обеспечивают совместимость по данным и организацию обмена данными между суперкомпьютером кластерного типа и системой автоматизации проектирования и позволяют избежать утечки информации о назначении и характеристиках проектируемого устройства. Решение последней проблемы состоит в том, чтобы предоставлять в веб-центр только необходимую для оптимизации проектных решений информацию, что достигается путем преобразования исходных данных и оптимизационной задачи к абстрактному виду, в которых скрыта информация о функциональности проектируемого устройства. Описываемая технология была апробирована при в рамках системы автоматизации логического проектирования заказных КМОП СБИС на основе

удаленного решения задачи нахождения корней логических уравнений на базе суперкомпьютера [13].

Список литературы

- [1] Поляков, А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры / А.К. Поляков. – М.: Солон-Пресс, 2003. – 320 с.
- [2] Сервер OpenSSH // [Электронный ресурс] https://help.ubuntu.ru/wiki/руководство_по_ubuntu_server/удаленное_администрирование/openssh_server. – Date of access: 10.11.2020.
- [3] Hauck, S. Data security for Web-based CAD / S. Hauck, S. Knol // Proceedings 1998 Design and Automation Conference. 35th DAC (Cat. No.98CH36175), San Francisco, CA, USA, 1998. – P. 788–793. Doi: 10.1145/277044.277242.
- [4] Бибило, П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П.Н. Бибило. – М.: СОЛОН-Пресс, 2005. – 384 с.
- [5] Операционный центр национальной грид-сети Республики Беларусь // – [Электронный ресурс] <http://noc.grid.by/index.php?n=Main.Download>. – Date of access: 10.11.2020.
- [6] Schuller, B. General Introduction to UNICORE // – [Электронный ресурс] <http://omega.sp.susu.ac.ru/books/conference/PaVT2013/talks/Tutorial-UNICORE-1-Schuller.pdf>. – Date of access: 10.11.2020.
- [7] Костельцев, А. В. Построение интерпретаторов и компиляторов: Использование программ BIZON, BYACC, ZUBR / А.В.Костельцев. – Санкт-Петербург: Наука и техника, 2001. – 224 с.
- [8] Rubin, S.M. Computer Aids for VLSI Design / S.M. Rubin. – Addison-Wesley Publishing Company, 1987. – 458 p.
- [9] 6 IEEE Standard for VHDL Register Transfer Level (RTL) Synthesis // – [Электронный ресурс] http://www.ece.gatech.edu/academic/courses/spring2007/ece4170/DesignDocumentation/IEEE_%206.pdf. – Date of access: 10.11.2020.
- [10] Бибило, П.Н. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний / П.Н. Бибило, В.И. Романов. – Минск: Беларус. навука, 2011. – 279 с.
- [11] Sentovich, E.M. SIS: A system for sequential circuit synthesis. Technical Report UCB/ERL M92/41, EECS Department / E.M. Sentovich, K.J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P.R. Stephan, R.K. Brayton, A.L. Sangiovanni-Vincentelli. – University of California: Berkeley, 1992.
- [12] Черемисинов, Д.И. Анализ и преобразование структурных описаний СБИС / Д.И. Черемисинов. – Минск: Беларус. навука, 2006. – 275 с.
- [13] Черемисинов, Д. И. Использование параллельных вычислений при автоматизированном проектировании СБИС / Д. И. Черемисинов, Л. Д. Черемисина // Проблемы разработки перспективных микро- и наноэлектронных систем. Сборник трудов / под общ. ред. академика РАН А. Л. Стемпковского, М.: ИППМ РАН, 2016. Часть I. – С. 32-39.

USING GRID SYSTEMS TO PERFORM TIME-CONSUMING VLSI DESIGN OPERATIONS

D.I. CHEREMISINOV

*Leading researcher of UIIP of NAS of Belarus,
candidate of technical sciences, associate professor*

L.D. CHEREMISINOVA

*Principal researcher of UIIP of NAS of Belarus,
doctor of technical sciences, professor*

*United Institute of Informatics Problems of National Academy of Sciences of Belarus, Republic of Belarus
{cher, cld}@newman.bas-net.by*

Abstract. The problem of accelerating labor-intensive calculations in solving combinatorial problems of VLSI design by performing distributed calculations on remote supercomputers is considered. The methodology for using the grid system service to perform time-consuming calculations is described, which is based on open standards and middleware that is an agent between programs running on remote computers. Software tools are proposed that ensure data compatibility and the organization of data exchange between the grid system and the client design automation system, allowing to avoid leakage of information about the purpose and characteristics of the designed device.

Keywords: grid system, distributed computing, VLSI design automation, description format converters.