РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я
№ 6 1994

УДК 621.3.049.77 : 658.512.011.56

А. В. Пичуев, А. Г. Рябченко, Д. Г. Титов, С. А. Фролов
(Новосибирск)

О ПРОЕКТИРОВАНИИ СБИС ВЫСОКОСКОРОСТНОГО КРИПТОПРОЦЕССОРА

Рассматриваются вопросы проектирования СБИС, реализующей алгоритм криптографической обработки данных ГОСТ 28147-89, предназначенной для высокоскоростных сетей передачи данных с пропускной способностью до 10 Мбит/с. Обосновывается выбор архитектуры криптопроцессора, формируются требования к его основным узлам. Показывается эффективность системного подхода в процессе проектирования, приведен ряд универсальных решений, использованных для получения высокой производительности СБИС. Приводятся результаты испытаний спроектированного криптопроцессора.

Введение. В современных многопользовательских операционных системах, базах данных и телекоммуникационных сетях системы разграничения доступа и защиты информации являются составной частью и имеют важное значение. Во многих применениях (например, коммерческих или военных) требуется обеспечить гарантированную стойкость алгоритмов шифрования для передаваемой по каналам связи или хранимой информации. При реализации алгоритмов программным или аппаратным способом на универсальной элементной базе остается невысокой защищенность от считывания ключей шифрования, таблиц замены и других секретных данных, используемых в процессе шифрования. Поэтому разрабатываются специализированные СБИС, которые позволяют аппаратным способом обеспечить гарантированную недоступность ключевой информации для считывания из криптографической аппаратуры после ее инициализации и высокую пропускную способность, достаточную для того, чтобы вести криптографическую обработку в современных сетях передачи данных в реальном времени (например, в сети Ethernet, пропускная способность которой 10 Мбит/с).

Имеется ряд микросхем, реализующих алгоритм DES [1] и обеспечивающих различную пропускную способность, но применение их в отечественной аппаратуре затруднено в силу существующего законодательства США, регламентирующего экспорт этих изделий. К тому же алгоритм DES, разработанный в 1973 году, не удовлетворяет современным требованиям криптостойкости из-за малой длины ключа (всего 56 бит). Отечественный алгоритм ГОСТ 28147-89, принятый в 1989 году [2], лишен этого недостатка (длина ключа 256 бит) и отличается большей вычислительной сложностью, обеспечивающей гарантированную стойкость алгоритма при защите информации.

Существующая отечественная микросхема «Блюминг-1», реализующая ГОСТ 28147-89 (далее алгоритм), обеспечивает пропускную способность 200 Кбайт/с, что недостаточно для современных применений.

В данной статье описывается ряд универсальных решений, позволивших получить достаточно высокую производительность СБИС криптопроцессора

(2 Мбайт/с), а также методология проектирования схем управления трактом обработки данных.

I. **Разработка архитектуры СБИС.** При разработке архитектуры СБИС принимались во внимание в первую очередь следующие факторы:

1. Стандартом на алгоритм предусмотрены три режима шифрования данных и режим выработки имитовставки (фактически зашифрованной «контрольной суммы» сообщения). СБИС должна реализовать все эти режимы, а также команды загрузки ключевой информации.

2. Так как предполагалось использовать СБИС во многих приложениях, то ее интерфейс должен по возможности быть простым и универсальным.

3. По алгоритму кодирование/декодирование информации производится блоками по 64 бита, поэтому, если ввод/вывод данных осуществлять через шину меньшей разрядности, то шифрование начнется только после накопления 64 бит данных, что может приводить к простоям в тракте обработки данных.

4. СБИС должна обеспечивать производительность не менее 10 Мбит/с для использования ее при создании криптографической аппаратуры как для глобальных, так и для локальных сетей.

Исходя из п. 1 была выбрана архитектура СБИС с микропрограммным управлением, что позволило перенести сложность реализации всех режимов работы СБИС из аппаратной в программную область. Таким образом, работа по проектированию блока управления СБИС была разделена на более простые этапы: во-первых, разработка универсальной схемы управления с памятью микропрограмм и оптимизация ее при моделировании на простых программах, во-вторых, отладка сложных программ, реализующих управление шифрованием и уточнение архитектуры тракта обработки данных и блока микропрограммного управления.

При обсуждении п. 2 было принято, что определение точной спецификации выводов СБИС не должно сдерживать или влиять на проектирование основных блоков микросхемы. Поэтому все узлы, относящиеся к внешним сигналам управления, были выделены в отдельный интерфейсный блок и выработана спецификация для внутренних сигналов связи с интерфейсным блоком. Это позволило вести разработку интерфейсного блока и совершенствование внешнего управления СБИС на протяжении всего этапа проектирования криптопроцессора.

Что касается п. 3, то для оптимизации пропускной способности СБИС были введены входной и выходной буферы так, чтобы запись во входной буфер/чтение из выходного буфера и обработка данных могли бы проводиться независимо друг от друга, т. е. ввод/вывод и обработка данных осуществлялись бы конвейеризировано.

На этапе проектирования архитектуры СБИС управление вводом во входной буфер и выводом из выходного буфера было организовано специальным параметризуемым блоком (переменной является размер буферов) для того, чтобы определить оптимальную емкость буферов на этапе моделирования всей системы в целом. При этом с изменением емкости буферов меняется лишь небольшой блок СБИС без влияния на остальные блоки. Этим конфигурируемым блоком и стал интерфейсный блок. Требование одновременно с вводом/выводом проводить обработку данных привело к тому, что интерфейсный блок и блок микропрограммного управления должны работать асинхронно и сигнализировать друг другу о подготовленности данных с помощью специальных флагов. Доводы, связанные с обсуждением конструкции буферов, также повлияли на принятие решения при обсуждении п. 2. Четкое выделение интерфейсного блока в отдельный модуль как схемотехнически, так и топологически позволяет изготавливать модификации СБИС как с 8-разрядной двунаправленной шиной данных, так и с 16- и 32-разрядной одно- или двунаправленными шинами данных, изменяя лишь часть интерфейсного блока.

Согласно п. 4 требуется оптимизация архитектуры тракта обработки данных с учетом специфических вычислений, используемых в алгоритме, поэто-

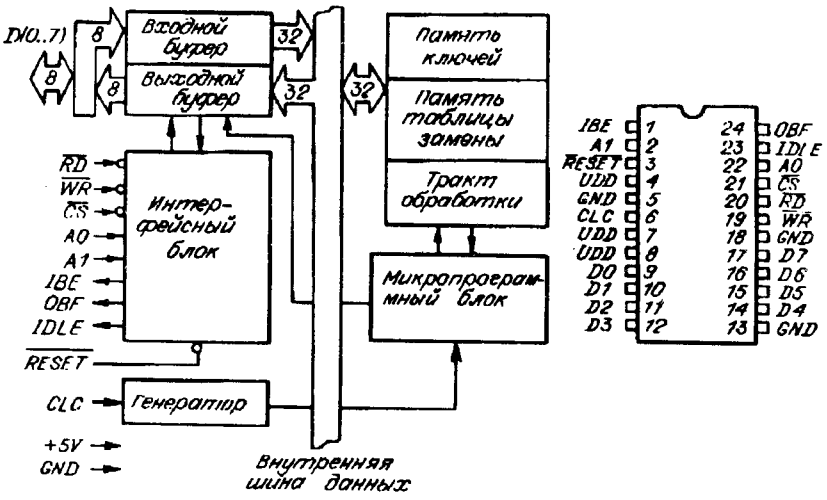

Рис. 1. Структурная схема криптопроцессора и конфигурация выводов

му этот блок не является универсальным. Структурная схема СБИС показана на рис. 1.

Функционально СБИС представляет собой конвейер с тремя ступенями передачи данных: входной буфер, тракт обработки данных и выходной буфер. Конвейер управляется блоком микропрограммного управления и интерфейсным блоком. Интерфейсный блок управляет загрузкой данных во входной буфер и выгрузкой данных из выходного буфера. Микропрограммный блок вырабатывает сигналы для считывания данных из первой ступени конвейера (входной буфер) во вторую ступень (тракт обработки) и из второй ступени в третью (выходной буфер). Внешние сигналы СБИС IBE («Входной буфер пуст»), OBF («Выходной буфер полон») и IDLE («Тракт обработки данных свободен») характеризуют состояние трех ступеней конвейера в каждый момент времени.

Подобная организация передачи данных позволяет оптимизировать общую скорость обработки данных в микросхеме (или «передачу данных сквозь СБИС»), поскольку с точки зрения внешнего управления криптопроцессор представляет собой структуру типа FIFO («первым вошел — первым вышел») с емкостью 24 байта (8 байт входного буфера плюс 8 байт, находящихся в тракте обработки, плюс 8 байт выходного буфера).

II. Блок микропрограммного управления (БМУ). БМУ состоит из памяти микропрограмм и схемы выбора адреса очередного слова микропрограммы. Схема выбора адреса содержит стек, счетчики и мультиплексор, позволяющий выбирать адрес из нескольких источников (рис. 2).

Создаваемые микропрограммы могут оперировать такими языковыми конструкциями, как:

«если ⟨флаг установлен⟩, то ⟨переход по адресу⟩»,
«безусловный переход по адресу»,
«вызов подпрограммы»,
«выход из подпрограммы»,
«инициализация счетчика»,
«декрементирование счетчика».

Для удобства использования этих конструкций был разработан специальный язык ассемблерного уровня. Текст программы на этом языке может транслироваться как в описание модели памяти программ на языке HHDL [3], так и в таблицу для генерации топологии памяти микропрограмм.

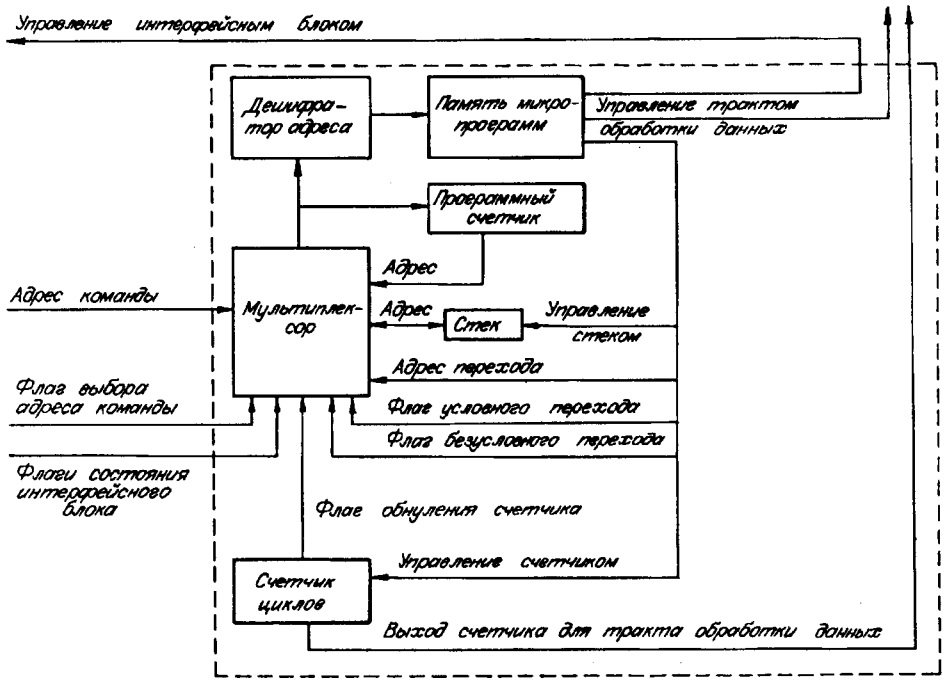

Рис. 2. Структурная схема блока микропрограммного управления

Приведем пример, иллюстрирующий взаимодействие блока микропрограммного управления и интерфейсного блока (ИБ) на примере программы по загрузке 256-битного ключа шифрования.

После поступления команды по загрузке ключа шифрования БМУ инициализирует внутренний счетчик, значение которого используется в качестве адреса памяти ключа шифрования, организованной как восемь 32-разрядных слов. ИБ независимо от БМУ загружает входной буфер, имеющий емкость 64 бита и состоящий из двух (младшей и старшей) 32-битных частей (см. разд. I), после чего устанавливает флаг flag_IBF (входной буфер заполнен). Затем содержимое младшей части буфера через 32-разрядную шину тракта обработки данных записывается в память ключей и содержимое счетчика декрементируется. Далее считывается содержимое старшей части буфера и содержимое счетчика декрементируется снова. Блок микропрограммного управления вырабатывает сигнал mIBE (входной буфер пуст) для ИБ, и процесс записи ключа во входной буфер продолжается. Этот процесс повторяется 4 раза, после чего вырабатывается флаг обнуления счетчика и происходит условный переход на цикл ожидания следующей команды.

Текст микропрограммы выглядит следующим образом (каждое микропрограммное слово заключается в скобки [ ]):

```
        // инициализация счетчика
  [ CntLoad; NewCntValue = 7; ]
        // цикл ожидания заполнения входного буфера
waitWK: [ IF_NOT; FlagSel = flag_IBF; GotoAdr = waitWK; ]
        // запись части ключа из младшей части буфера
  [ enable_IBL; key_WRITE; ]
        // декрементирование счетчика
  [ Cnt_Decrement; ]
        // запись части ключа из старшей части буфера
```

```
[ enable_IBH; key_WRITE; mIBE;
    // входной буфер опустошен
    // условный переход в случае обнуления счетчика
IF; FlagSel = flag_Zero; GotoAdr = WAIT_LOOP; ]
    // безусловный переход на цикл ожидания
    // заполнение входного буфера и
    // декрементирование счетчика
[ Goto; GotoAdr = waitWK; Cnt_Decrement; ]

WAIT_LOOP: [ mIDLE; Goto; GotoAdr = WAIT_LOOP; ]
```

Схема блока микропрограммного управления и конструкции описанного языка программирования представляют собой достаточно универсальную систему, которая может быть использована при проектировании микросхем с другим назначением. При этом, если требуется, можно переопределить имена управляющих сигналов, изменить количество счетчиков и порядок счета, емкость стека, количество и назначение флагов.

III. **Тракт обработки данных.** Как было отмечено ранее, алгоритм предусматривает четыре режима работы с данными. Все они базируются на одном наборе операций с небольшими вариациями. На рис. 3 показана последовательность преобразования данных, которая является общей для всех четырех режимов и повторяется от 16 до 32 раз для одного блока данных в зависимости от производимой операции, при этом выполняются следующие операции:

— выборка ключа из памяти;

— сложение по модулю 2^{32} с накопленным в предыдущем такте результатом;

— выборка значения из таблицы замены по адресу, полученному в результате сложения;

— циклический сдвиг;

— операция «Исключающее ИЛИ» с накопленным в предыдущем такте результатом.

Пунктирной линией на рисунке показана пересылка данных при выполнении одного из 32 циклов алгоритма. При первом цикле шифрования данные загружаются из входного потока блоками по 8 байт. В последующих циклах результаты обработки содержимого регистров *A* и *B* записываются в регистр *C*, затем содержимое *A* переписывается в *B*, после чего *C* переписывается в *A* и начинается выполнение следующего цикла.

Для повышения быстродействия микросхемы был применен специализированный тракт обработки данных, позволяющий производить всю эту цепочку вычислений за один микропрограммный такт (при использовании 2 мкм КМОП-технологии время такта составило 80 нс). Это стало возможным благодаря конвейеризации и применению специальных схем в реализации критичных по быстродействию узлов [4].

IV. **Интерфейсный блок.** При разработке интерфейсного блока основное внимание уделялось следующим требованиям:

— пропускная способность шины данных должна обеспечивать производительность криптопроцессора не менее 10 Мбит/с;

— интерфейс криптопроцессора должен предоставлять возможность работы по прерываниям;

— временные диаграммы сигналов должны быть совместимы с временными диаграммами существующих микроконтроллеров, используемых в ЛВС.

Для обеспечения производительности СБИС 10 Мбит/с шина данных должна пропускать 20 Мбит/с (чтение/запись). При взаимодействии криптопроцессора с аппаратурой сопряжения на каждую операцию чтения/записи в СБИС требуется произвести дополнительную операцию записи/чтения. Отсюда вытекает требование, чтобы шина пропускала 40 Мбит/с. Для использования криптопроцессора в системах с управляющим микропроцессором необходимо как минимум удвоить пропускную способность шины данных, чтобы было возможно производить дополнительные вычисления, т. е. СБИС должна пропускать данные со скоростью не менее 80 Мбит/с.

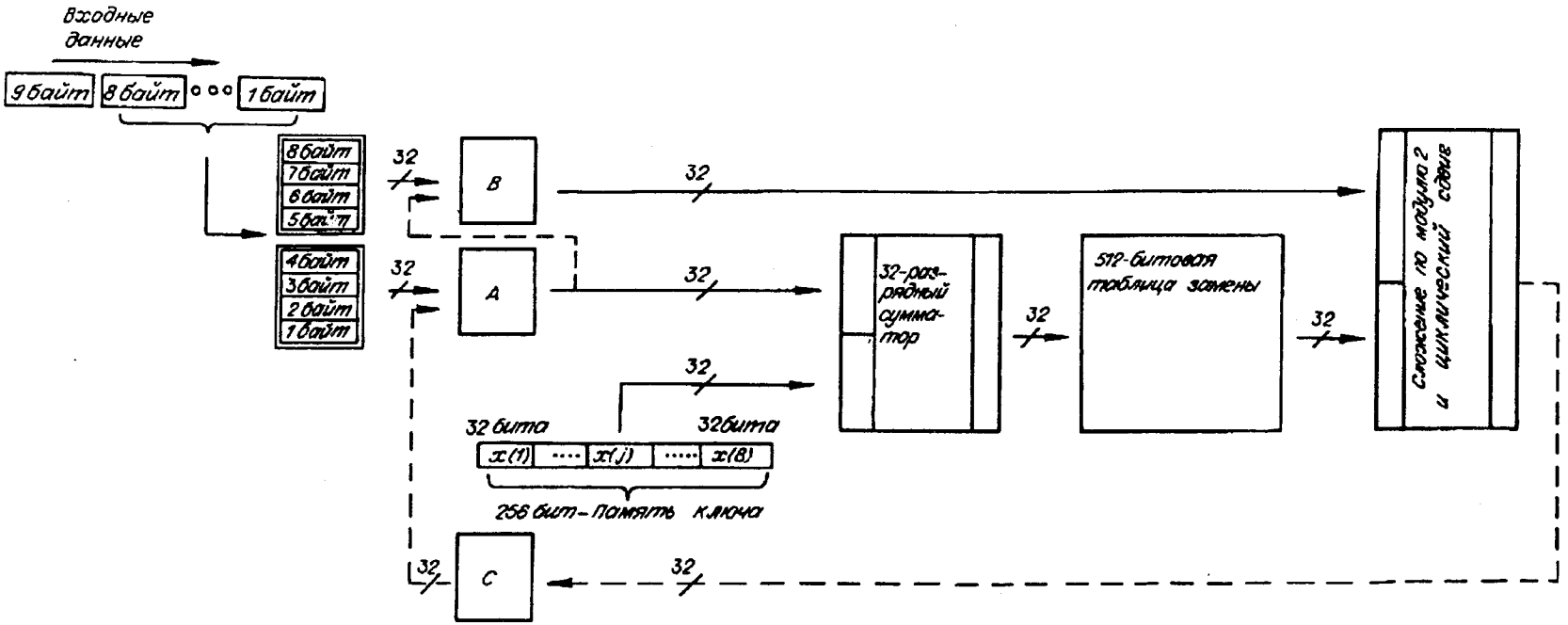

Рис. 3. Шифрование данных в режиме простой замены

Так как на сегодня наиболее распространены 8-разрядные микроконтроллеры, то был выбран 8-разрядный интерфейс с одной двунаправленной шиной данных и пропускной способностью 10 Мбайт/с.

Для возможности эффективного использования криптопроцессора в системах реального времени могут быть использованы три выходных сигнала СБИС: «Входной буфер пуст», «Выходной буфер полон», «Внутренний конвейер пуст». Прерывания по этим трем сигналам позволяют обеспечить полное управление криптопроцессором.

При составлении временных диаграмм за основу были взяты диаграммы микроконтроллеров семейства MSC-51. Принятие такого решения обусловлено широким распространением этих микроконтроллеров в мире, а также наличием отечественных аналогов. Структурная схема криптопроцессора и конфигурация выводов приведены на рис. 1.

V. **Система межблочной синхронизации.** Для выработки синхроимпульсов на СБИС подается тактирующий сигнал CLC. Система синхронизации СБИС криптопроцессора построена на принципе двухфазной синхронизации, описанной в [5]. За счет временного разноса между активными уровнями синхроимпульсов Ф1 и Ф2 удается избежать эффекта «гонок», но теряется производительность, так как это время фактически не используется при работе СБИС. Время разноса T вычисляется при моделировании и затем, обычно с большим запасом, учитывается при проектировании схемы генератора двухфазной синхронизации. Чтобы минимизировать это время, а следовательно, повысить быстродействие СБИС, была разработана схема генератора четырехфазной синхронизации, в которой нет задержки между вырабатываемыми синхроимпульсами Ф1, Ф2, Ф3, Ф4.

В такой «псевдочетырехфазной» системе синхронизации синхроимпульсы Ф1 и Ф3 аналогичны сигналам Ф1 и Ф2 двухфазной синхронизации, а сигналы Ф2 и Ф4 — времени задержки T. Таким образом, регулируя во внешней схеме скважность сигнала CLC, можно экспериментально найти минимальное время длительности сигналов Ф2 и Ф4, при котором СБИС еще работоспособна. При этом производительность СБИС повышается за счет снижения времени разноса между синхроимпульсами Ф1 и Ф3, которое в случае системы двухфазной синхронизации выбирается с относительно большим запасом и является фиксированным в СБИС после ее изготовления.

VI. **Реализация.** Микросхема криптопроцессора была спроектирована с использованием собственных средств проектирования [6—12] по 2-микронной КМОП-технологии с одним уровнем металлизации. Размер кристалла составил 5,5 × 5,8 мм при сложности около 50 тыс. транзисторов. Потребление СБИС 10 мВт. Полученные экспериментальные образцы работали на частотах до 20 МГц (при скважности тактирующего сигнала 0,7), что обеспечивает производительность до 2 Мбайт/с. Благодаря использованию единого программного комплекса и современной методологии проектирования работоспособные образцы были получены с первой производственной итерации.

Заключение. Использование описанных подходов к проектированию архитектуры и узлов позволило разработать микросхему для создания криптографической аппаратуры, способной закрывать высокоскоростные сети передачи данных, такие как Ethernet и X.25. При разработке криптопроцессора были усовершенствованы имеющиеся средства проектирования и объединены в единый программный комплекс, а также отработана применяемая методология проектирования.

Авторы благодарят А. А. Симонова, В. В. Чеснокова, Д. Г. Фризена, К. К. Смирнова, С. Б. Рахимова, а также технологов НПО «Восток» за содействие при проведении работ по созданию криптопроцессора.

СПИСОК ЛИТЕРАТУРЫ

1. Data Encryption Standard: Federal Information Processing Standard 46.
2. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.—М.: Госстандарт, 1989.
3. HHDL Reference Manual.— Silvar-Lisco, 1983.

4. Weste N., Eshraghian K. Principles of CMOS VLSI Design: A Systems Perspective.—Reading: Addison-Wesley, 1989.—P. 311.
5. Meed C., Convey L. Introduction to VLSI Systems.—Reading: Addison-Wesley, 1980.
6. Лившиц З. А., Пичуев А. В. SimSim: программа логического моделирования МОП БИС на переключательном уровне // Автометрия.—1991.—№ 3.
7. Лившиц З. А., Смирнов К. К. LOGIC — программный комплекс для генерации топологии ПЛМ // Автометрия.—1991.—№ 1.
8. Лившиц З. А., Титов Д. Г. Алгоритмы работы с тайловыми представлениями топологии СБИС // Там же.
9. Рябченко А. Г. MICE2: программа экстракции электрической схемы из описания топологии МОП СБИС // Автометрия.—1991.—№ 5.
10. Пичуева Ю. А. Алгоритм ветвления для решения задачи свертывания программируемых логических матриц // Автометрия.—1992.—№ 5.
11. Титов Д. Г. Система проектирования топологии интегральных схем ICE.—Новосибирск, 1989.—(Препр. /ИАиЭ СО АН СССР; 464).
12. Титов Д. Г. Алгоритмы иерархической верификации геометрических ограничений на топологию СБИС // Автометрия.—1994.—№ 6.

Поступила в редакцию 26 мая 1994 г.

Реклама продукции в нашем журнале — залог Вашего успеха!