

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 519.7

П.Н. Бибило, С.Н. Кардаш, В.И. Романов

**СиВер – СИСТЕМА СИНТЕЗА И ВЕРИФИКАЦИИ
КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ СХЕМ**

Описывается система синтеза и верификации многовыходных комбинационных схем в базе библиотек проектирования базовых матричных кристаллов. Исходными данными являются матричные и скобочные описания систем булевых функций, а также описания на языке VHDL. Система СиВер состыкована с системой синтеза LeonardoSpectrum и имеет программы логической минимизации и верификации, отсутствующие в LeonardoSpectrum. Совместное использование СиВер и LeonardoSpectrum позволяет получать логические схемы, имеющие меньшую сложность и задержку.

Введение

Создание систем автоматизированного проектирования сверхбольших интегральных схем (СБИС) представляет собой сложную наукоемкую проблему. Одним из типов цифровых СБИС являются базовые матричные кристаллы (БМК) [1], относящиеся к классу полузаказных СБИС. Степень интеграции БМК постоянно возрастает, и они по многим технико-экономическим показателям по-прежнему остаются подходящей элементной базой для многих применений.

В проектировании БМК можно выделить логический и топологический этапы проектирования. На этапе логического проектирования решаются задачи синтеза логических схем, размещаемых на БМК, моделирования и верификации. Задача синтеза логических схем БМК ставится как задача синтеза в заданной библиотеке логических элементов – библиотеке проектирования, называемой также целевой библиотекой синтеза или просто библиотекой БМК. Библиотека БМК может включать достаточно обширную, но ограниченную номенклатуру элементов, позволяющих реализовывать на БМК разнообразные микроэлектронные устройства. Так, например, в состав библиотеки БМК серии K1574 [2] входят около 80 функциональных элементов различных типов: комбинационные элементы, базовые вентили, триггеры, мультиплексоры и т. п. Элемент библиотеки БМК представляется в виде сети транзисторов и характеризуется «сложностью» – числом элементарных ячеек БМК, необходимых для его размещения на кристалле. Ячейка БМК содержит пару транзисторов, соотношение размеров которых оптимизировано с целью минимизации площади кристалла и повышения его быстродействия. Обычно ячейки БМК располагаются в линейках, в которых размещаются элементы логических схем, в каналах между линейками осуществляется трассировка соединений. Размещение элементов и трассировка соединений – основные задачи топологического проектирования БМК.

При проектировании схем на БМК одной из основных целей оптимизации является минимизация площади, занимаемой схемой на кристалле. Этап логического проектирования в этом смысле является решающим: получение «хорошей» логической схемы позволяет в значительной степени минимизировать площадь кристалла в целом.

Этап топологического проектирования решается развитыми средствами САПР фирм Cadence, Mentor Graphics и др. Для решения задач логического проектирования БМК могут применяться система синтеза логических схем LeonardoSpectrum и система моделирования ModelSim, разработанные фирмой Mentor Graphics. Однако в этих системах отсутствуют средства полной логической верификации, синтез логических схем ведется от исходных алгоритмических описаний на языке VHDL [3] и ориентирован на описания общего вида [4].

Предлагаемая в настоящей работе система СиВер ориентирована на синтез комбинационных логических схем. В составе системы имеются развитые средства логической минимизации систем булевых функций в классе ДНФ и программы полной логической верификации

схем. Система СиВер полностью состыкована по языку VHDL с системой синтеза (синтезатором) LeonardoSpectrum и системой моделирования ModelSim. Совместное использование СиВер и LeonardoSpectrum позволяет во многих случаях получать более эффективные схемные решения и верифицировать проектные действия на этапе логического проектирования.

1. Общая характеристика системы

Система СиВер предназначена для синтеза комбинационных логических схем в библиотеке проектирования БМК и верификации таких схем. Исходными данными системы являются:

- функционально-структурные описания логических схем на языке VHDL;
- RTL-описания логических схем, являющиеся промежуточными VHDL-описаниями (данными) [3] в результате высокоуровневого синтеза в LeonardoSpectrum;
- структурные описания (netlist) логических схем, являющиеся результирующими данными после синтеза в LeonardoSpectrum и представленные в формате VHDL (стандартном обменном формате зарубежных САПР);
- матричные и скобочные представления систем полностью определенных булевых функций на языке SF [5].

Результат работы системы – структурные описания схем в базе библиотечных элементов БМК серии K1574, представленные в формате VHDL, который является исходным для топологического проектирования БМК.

В отличие от САПР «Синтез БМК» [6], ориентированной на структурный формат EDIF взаимодействия с другими САПР, в СиВер взаимодействие (обмен данными) осуществляется с использованием языка VHDL. Выбор в качестве интерфейсного языка VHDL [3, 4] не случаен. В настоящее время именно он получил наибольшее распространение, является международным стандартом и входным языком практически всех промышленных систем автоматизированного проектирования заказных, полузаказных и программируемых логических интегральных схем (ПЛИС).

Вместе с тем для осуществления функционально-структурного проектирования, связанного с проведением преобразований комбинационной части проектируемой схемы и логической минимизацией, оказалось полезным использование языка SF [5], близкого к иерархическому представлению RTL-описаний. Схема технологического взаимодействия системы СиВер с «внешним миром», в роли которого чаще всего выступает САПР LeonardoSpectrum, показана на рис. 1. На этом рисунке блоки CSF и FormatSF представляют собой специальные программы, осуществляющие конвертирование описаний проектируемых схем с языка VHDL на язык SF и обратно.

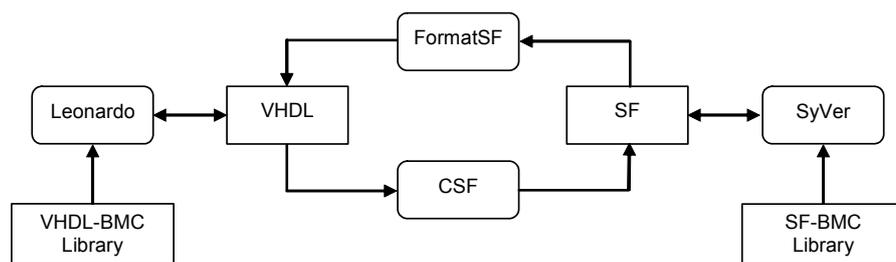


Рис.1. Общая схема технологического взаимодействия систем СиВер и LeonardoSpectrum

Взаимодействие систем СиВер и LeonardoSpectrum по терминологии работы [7] представляет собой пример стыковки горизонтальных САПР на этапе логического проектирования. Если проектировщик начинает проектирование в LeonardoSpectrum, то исходными данными могут быть любые алгоритмические VHDL-описания комбинационных схем, так как в таких VHDL-описаниях после этапа высокоуровневого синтеза имеются только логические операторы.

2. Типовой технологический маршрут проектирования

При использовании системы СиВер типовой маршрут проектирования может быть описан следующим набором проектных действий:

1. Описание проектируемой схемы, представленное на языке VHDL при помощи команды unpar САПР Leonardo [5], строится в виде единого модуля, подаваемого на вход конвертора CSF [8]. Результат конвертирования – пара файлов, содержащих описание той же схемы на языке SF. В файле с именем *ИМЯ_СХЕМЫ.SF* содержится функциональное описание схемы, в то время как файл с именем *ИМЯ_СХЕМЫ_CON.SF* представляет ее структурное описание.

2. Полученные на предыдущем этапе файлы служат основой для создания нового проекта в рамках системы СиВер, который подвергается некоторому набору оптимизационных преобразований. Вначале в целях уменьшения сложности функциональное описание синтезируемого устройства модифицируется при помощи методов минимизации и декомпозиции. Сложность описания обычно оценивается суммарным числом литералов всех дизъюнктивных нормальных форм, задающих функционирование устройства. По завершении этих преобразований строится покрытие рассматриваемой схемы элементами используемой БМК. В результате полученное SF-описание схемы отражает двухуровневую иерархию: в корневом узле содержится полное структурное описание схемы, а листовые элементы соответствуют тем или иным элементам БМК.

3. При помощи конвертора FormatSF [9], входящего в состав системы СиВер, описание структуры схемы из головного SF-блока проекта конвертируется в модуль на языке VHDL. Совместно с VHDL-компонентой БМК, содержащей описания отдельных элементов библиотеки, данный модуль задает проектируемую схему, которая далее может быть передана на моделирование в систему ModelSim, повторный синтез в системе LeonardoSpectrum или топологическое проектирование, выполняемые соответствующими программными комплексами.

4. В целях обеспечения высокого уровня надежности проектирования описание полученной схемы может быть верифицировано. Для этого снова осуществляется конвертирование описания в язык SF и после построения нового проекта реализуется его верификация с заключительным состоянием первичного проекта, подвергавшегося обработке в системе СиВер согласно п. 2.

3. Функциональные возможности системы СиВер

В системе поддерживаются следующие этапы процесса функционально-структурного проектирования комбинационного устройства:

- конвертирование функционально-структурных описаний проектируемого устройства с языка высокого уровня VHDL на язык SF и обратно;
- выбор типов библиотечных элементов для реализации комбинационной части устройства (формирование библиотеки БМК);
- оптимизация функционального описания устройства;
- синтез логической схемы в заданной библиотеке БМК;
- логическая верификация описания функционирования устройства.

Для реализации указанных этапов разработан следующий набор проектных операций.

1. Устранение иерархии. Иерархическое описание схемы основывается на выделении в схеме некоторых подсхем, присвоении этим подсхемам индивидуальных имен и описании схемы в целом в терминах соединения полюсов выделенных подсхем. В выделенной подсхеме можно опять выделять свои подсхемы, так появляется еще один уровень в иерархии описания схемы в целом. Проектная операция Устранение иерархии имеет своей целью устранение условных границ, выделяющих подсхемы. Процедура преобразует иерархическое SF-описание произвольной глубины вложенности в одноуровневое SF-описание, используя функционально-структурные описания составляющих его блоков. Одноуровневое (по числу уровней иерархии) функциональное описание содержит логические уравнения или систему ДНФ, задающие функционирование логической схемы. При выполнении данной операции производится исключение уравнений связи типа $y1=x1$. Соответствующие алгоритмы и программы описаны в работе [10].

2. Устранение иерархии без редукции уравнений связи. Эта проектная операция подобна операции Устранение иерархии, за исключением того, что после ее выполнения в системе логических уравнений остаются уравнения связи типа $y1=x1$.

3. Ликвидация внутренних переменных. Назначение данной операции – представить функции выделенной комбинационной части схемы в виде системы дизъюнктивных нормаль-

ных форм (ДНФ) булевых функций. В языке SF система ДНФ булевых функций описывается парой матриц – троичной и булевой. Троичная матрица задает элементарные конъюнкции, которые входят в ДНФ функций системы, единичные элементы булевой матрицы отмечают факты вхождения конъюнкций в ДНФ функций. Алгоритмы выполнения данной операции описаны в работе [11].

4. Факторизация. Данная операция применяется к системам ДНФ или логических уравнений. Ее результатом является система «ограниченных» логических уравнений: ранги (число литералов) всех конъюнкций и дизъюнкций не превышают заданного числа, диктуемого используемой библиотекой примитивов БМК. Эта система задает, по сути, многовыходную логическую сеть в технологически независимом базисе операторов И, ИЛИ, НЕ. Использование этой операции перед операцией Покрытие позволяет оптимизировать результирующую логическую сеть за счет выделения и однократной реализации общих частей логических уравнений. Алгоритмы выполнения операции факторизации приведены в работе [12].

5. Раздельная минимизация системы булевых функций. Данная операция производит раздельную минимизацию в классе ДНФ функций комбинационной части схемы. Методы и алгоритмы выполнения данной операции представлены в работе [13].

6. Совместная минимизация системы булевых функций. Данная операция производит совместную минимизацию в классе ДНФ функций комбинационной части схемы. Методы и алгоритмы выполнения данной операции представлены в работе [13].

7. Покрытие сети элементами БМК. Данная операция предназначена для получения комбинационной логической схемы в базисе элементов заданной библиотеки БМК. Исходными данными является система логических уравнений в базисе многоместных операторов И, ИЛИ и одноместного оператора НЕ, результатом – многовыходная комбинационная схема в базисе элементов БМК. Логические уравнения при выполнении проектной операции Покрытие сети элементами БМК интерпретируются как ациклические булевы сети, которые затем покрываются библиотечными элементами. Для этого используется эвристический алгоритм, который ориентируется на построение логических схем из библиотечных элементов, в совокупности занимающих минимальное число ячеек на кристалле.

Действия, производимые при синтезе, выполняются в следующей последовательности:

- 1) исходное описание представляется булевой сетью;
- 2) булева сеть разбивается на несколько подсетей, каждая из которых имеет только одну выходную вершину;
- 3) каждая подсеть покрывается библиотечными элементами.

Булева сеть – это ориентированный ациклический граф. Вершины графа, обладающие нулевой полустепенью исхода, объявляются как выходы сети, а вершины с нулевой полустепенью захода – как входы.

Каждой вершине графа соответствует некоторая переменная. Переменные, соответствующие входам сети, называются входными, а выходам сети – выходными. Переменные, соответствующие остальным вершинам сети, называются промежуточными. Вершина a_i является входной для вершины a_j , если имеется дуга, направленная из a_i в a_j .

Таким образом, булева сеть является удобной моделью для представления системы логических уравнений в базисе многоместных операторов И, ИЛИ и одноместного оператора НЕ. Построение булевой сети по алгебраическому выражению, задающему логическое уравнение, ведется следующим образом. Для каждого знака логической операции выражения строится вершина сети и ей в соответствие ставятся данная операция и некоторая промежуточная переменная. К полученному множеству вершин добавляются входные вершины, проводятся необходимые соединения дугами.

С другой стороны, булева сеть может служить для представления комбинационной схемы из логических элементов библиотеки БМК.

Логические уравнения, соответствующие библиотечным элементам, представляются в виде кластеров булевой сети. *Кластер* – это некоторая связанная подсеть булевой сети. Единственная вершина кластера, обладающая нулевой полустепенью исхода, помечается как выход, а вершины с нулевой полустепенью захода – как входы. Например, библиотечному элементу NO3A2, реализующему функцию $Y = \wedge(A + B + C * D)$, соответствует кластер, представлен-

ный на рис. 2. Каждый такой кластер будет характеризоваться стоимостью – числом базовых ячеек соответствующего библиотечного элемента. Например, для элемента NO3A2 это число равно пяти.

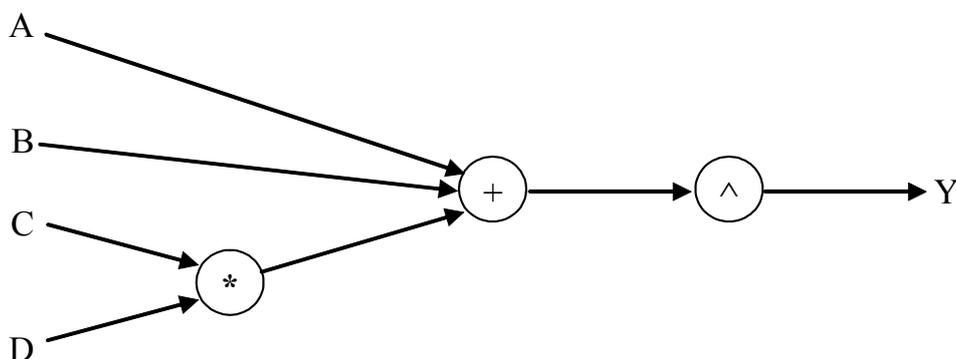


Рис. 2. Кластер, соответствующий библиотечному элементу NO3A2

Покрывающим элементом для некоторого кластера K булевой сети G называется библиотечный элемент, чей кластер изоморфен K .

Задача оптимального покрытия булевой сети библиотечными элементами сводится к поиску минимального по стоимости покрытия булевой сети кластерами библиотечных элементов. Для решения этой задачи используется программа, описанная в работах [14, 15].

4. Подсистема верификации

Назначение процедур верификации – автоматическая проверка отношения эквивалентности двух объектов. В качестве сравниваемых объектов могут выступать описания двух различных проектов или два описания, представляющие различные состояния одного и того же проекта. Результатом работы процедур является сообщение, констатирующее эквивалентность контролируемых описаний или содержащее сведения с указанием их различий, если таковые обнаруживаются.

Подход 1. Выполнение процедуры верификации сводится к последовательному преобразованию описаний и получению их комбинационных и «нераскрываемых» частей. Комбинационные части верифицируемых описаний представляются в виде двух систем ДНФ: D_1 и D_2 . Получение систем ДНФ базируется на проектной процедуре Ликвидация внутренних переменных [11], в свою очередь опирающейся на операцию инверсирования системы ДНФ. Затем выполняется проверка отношения эквивалентности полученных таким образом систем ДНФ D_1 и D_2 , которая сводится к проверке тавтологии ДНФ.

Подход 2. Первым шагом процедуры являются устранение иерархии и представление сравниваемых схем в виде системы логических уравнений, при этом каждое уравнение описывает поведение некоторого логического элемента. Затем задача решения полученной системы логических уравнений сводится к проверке выполнимости конъюнктивных нормальных форм (КНФ), которая получается путем представления каждого уравнения в виде КНФ и объединения построенных таким способом «маленьких» КНФ в одну «большую». Задача проверки выполнимости КНФ решается при помощи алгоритма, предложенного в работе [16]. Если сравниваемые схемы оказались неэквивалентными, то выдается список выходов, которые у этих схем функционально неэквивалентны, а также входной набор, вызывающий различные реакции схем на этих выходах.

Верификации могут подвергаться различные виды представления проектов: один проект может быть представлен на структурном уровне, другой – на функциональном, т. е. в виде RTL-описания либо матричного (скобочного) представления системы функций.

5. Особенности программной реализации

С точки зрения архитектуры программный комплекс СиВер основан на общем подходе, предложенном для организации подобных систем и изложенном в работах [3, 7]. Согласно этому подходу система состоит из ядра, реализующего проектные процедуры, и оболочки, в которой сосредоточены сервисные функции обслуживания пользователя системы (рис. 3).

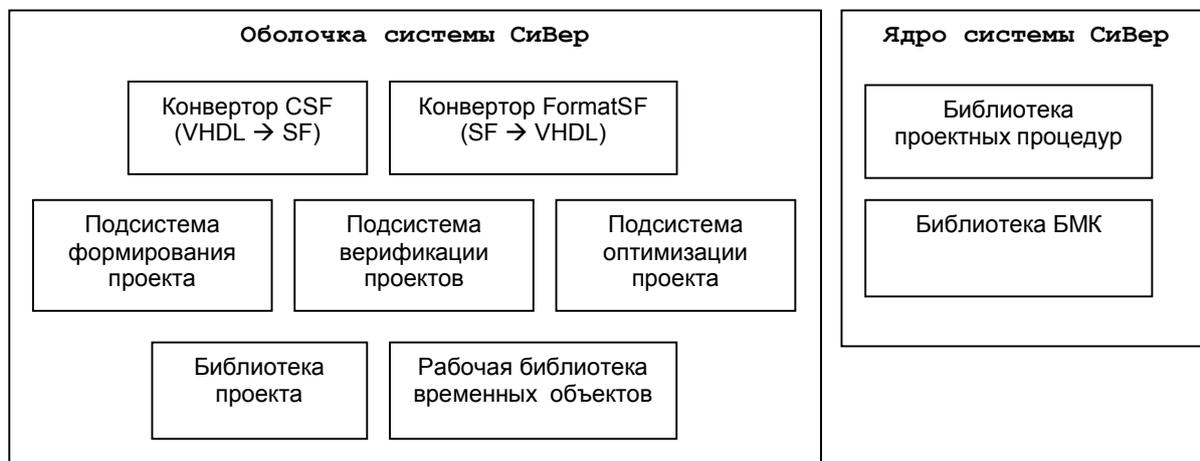


Рис. 3. Основные компоненты системы СиВер

Ядро системы СиВер включает в себя набор программных модулей, предназначенных для реализации всех проектных процедур, перечисленных при определении функциональных возможностей системы, – библиотеку проектных процедур, а также специальным образом организованную библиотеку матричных кристаллов.

Библиотека БМК состоит из отдельных именованных элементов. Каждый элемент имеет две формы представления – на языке SF и на языке VHDL. Кроме того, некоторые дополнительные параметры, характеризующие отдельный элемент библиотеки, представлены в виде именованной секции специального файла параметров элементов БМК. Организация этого файла выполнена по правилам организации файлов инициализации, определенных в операционной системе Windows.

Все элементы, описанные на языке SF, представляются в виде отдельных SF-файлов с именами, совпадающими с именами соответствующих элементов и имеющими расширение «.sf». Эти файлы должны располагаться в одной директории файловой системы. Путь доступа к этой директории является параметром системы СиВер и может задаваться при организации сеанса ее работы. Между сеансами этот путь сохраняется как значение ключа LIB_PATH в секции LIB_BMK файла инициализации сеанса работы – **siver.ini**. В дальнейшем определяемая таким способом директория будет называться директорией (каталогом) БМК.

Следует отметить, что вместе с SF-описаниями элементов в директории БМК должны присутствовать SF-файлы, хранящие CONNECT-настройки для подключения элемента. Такие файлы имеют имена вида **<name>_CON.sf**, где **<name>** – имя элемента.

Файл, содержащий описание дополнительных параметров элементов БМК, должен располагаться в директории БМК. Имя этого файла задается пользователем при развертывании сеанса и сохраняется в файле **siver.ini** как значение ключа LIB_PARAM в секции LIB_BMK.

В отличие от SF-описаний, представления элементов БМК на языке VHDL организуются в виде одного модуля VHDL, размещенного в директории БМК в виде одного файла. Имя этого файла задается пользователем при развертывании сеанса и сохраняется в файле **siver.ini** как значение ключа LIB_VHDL в секции LIB_BMK.

В отдельном сеансе работы системы СиВер пользователь должен определить библиотеку сеанса как подмножество элементов БМК, которые будут использоваться проектными операциями системы СиВер. Библиотека сеанса задается списком имен элементов (по одному в строке), который хранится в обычном текстовом файле. При организации сеанса пользователю пре-

доставляется возможность определения имени этого файла. Между сеансами путь доступа к текстовому файлу сохраняется как значение ключа LIB_LIST в секции LIB_BMK файла **siver.ini**.

Следует отметить, что при программной реализации системы СиВер особое внимание уделялось удобству пользователя. В частности, это связано с обеспечением динамического, настраиваемого пользователем меню проектных операций [17]. Внешние средства настройки этого меню позволяют пользователю самому определить как состав проектных операций, используемых на данном рабочем месте, так и их древовидную классификацию, наиболее точно отражающую содержание процесса проектирования для данного пользователя. Внешний вид рабочего окна системы показан на рис. 4.

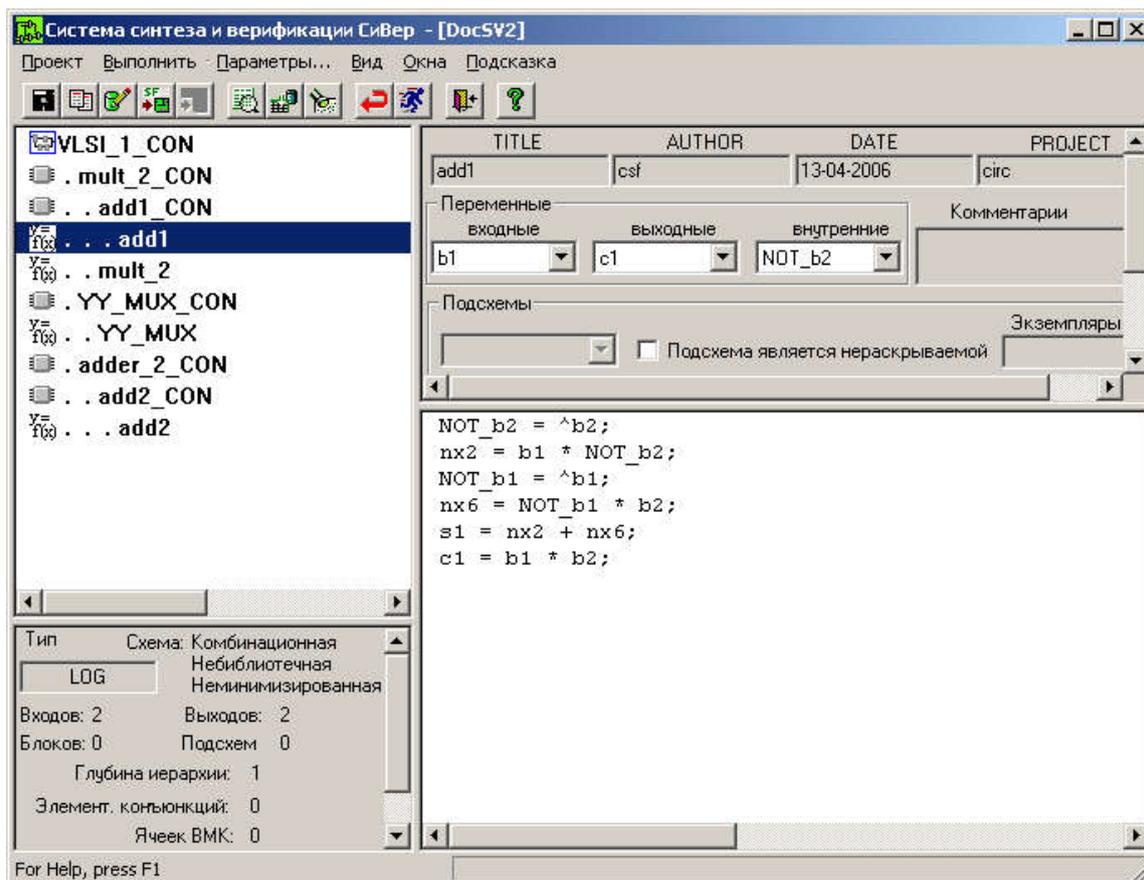


Рис. 4. Основное окно сеанса работы системы СиВер

6. Пример совместного использования систем СиВер и LeonardoSpectrum

Для иллюстрации возможностей, реализованных в системе СиВер, рассмотрим пример синтеза пятивходового дешифратора.

В листинге приведено алгоритмическое VHDL-описание дешифратора, имеющего пять информационных входов (*Din*), вход разрешения *enable* и 32 выхода (*Dout*). Оператор *s1* – это оператор сдвига влево, функция *TO_INTEGER* пакета *NUMERIC_STD* позволяет перейти от битового представления вектора *Din* к его представлению в виде целого числа (тип *INTEGER*).

Листинг. Алгоритмическое VHDL-описание дешифратора

```

library IEEE;
use IEEE.STD_LOGIC_1164. all;
use IEEE.NUMERIC_STD. all;

```

```

library exemplar;
use exemplar.exemplar_1164.all;

entity Decoder5_32 is
port (enable : in BIT;
      Din: in STD_LOGIC_VECTOR (4 downto 0);
      Dout: out STD_LOGIC_VECTOR (31 downto 0));
end;
architecture Synt of Decoder5_32 is
begin
with enable select
Dout <= sl ("00000000000000000000000000000001", TO_INTEGER (UNSIGNED(Din)))
      when '1',
      "00000000000000000000000000000000" when '0' ;
end Synt;

```

Синтез схемы от исходного алгоритмического описания, представленного в листинге, позволяет получить схему, суммарная площадь элементов которой (сложность схемы) составляет 321 элементарную ячейку БМК. Переход к RTL-описанию, конвертирование в SF-описание и получение системы ДНФ функций с последующими конвертацией в представление на языке VHDL и синтезом в LeonardoSpectrum приводит к схеме сложностью 150 элементарных ячеек. Это пример взаимодействия СиВер и LeonardoSpectrum с целью эффективного изменения формы представления исходных данных: алгоритмическое описание (листинг) было заменено VHDL-описанием соответствующей системой логических функций, представленных в ДНФ.

Заключение

Система СиВер по своим функциональным возможностям и алгоритмическому наполнению ориентирована на автоматизацию этапа логического синтеза СБИС. Размерности решаемых задач находятся в практическом диапазоне. Надежность проектирования обеспечивается программами верификации, сокращение площади кристалла – программами оптимизации, которые реализуют эффективные методы, проверенные экспериментально.

Система СиВер информационно состыкована с системой Custom Logic проектирования управляющей логики заказных СБИС [5] и может быть полезной для перевода проектов БМК на заказные СБИС.

Исследования проведены при частичной поддержке Белорусского республиканского фонда фундаментальных исследований (проект T05–258).

Список литературы

1. Быстродействующие матричные БИС и СБИС. Теория и проектирование / Б.Н. Файзулаев [и др.]; под общ. ред. Б.Н. Файзулаева и И.И. Шагурина. – М., 1989. – 304 с.
2. Лукошко, Г.К. КМОП – базовые матричные кристаллы серии K1574 / Г.К. Лукошко, Е.В. Коннов // Радиолюбитель. – 1997. – № 9. – С. 39–40.
3. Бибило, П.Н. Основы языка VHDL / П.Н. Бибило. – М.: СОЛОН-Р, 2002. – 224 с.
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. Система «Custom Logic» автоматизированного проектирования управляющей логики заказных цифровых СБИС / П.Н. Бибило [и др.] // Микроэлектроника. – 2004. – Т. 32. – № 5.
6. Система логического проектирования «Синтез БМК» / П.Н. Бибило [и др.] // УсиМ. – 2001. – № 3. – С. 28–35.
7. Бухтеев, А.В. Системы на кристалле. Новые тенденции / А.В. Бухтеев, В.В. Немудров // Электроника: наука, технологии, бизнес. – 2004. – № 3. – С. 52–56.
8. Кочанов, Д.А. Разработка лексического анализатора для транслятора VHDL-описаний логических схем / Д.А. Кочанов // Материалы Пятой Междунар. конф. «Computer-Aided Design

of Discrete Devices» (CAD DD'04), Минск, 15–17 ноября 2004 г. – Минск: ОИПИ НАН Беларуси, 2004. – Т. 2. – С. 179–186.

9. Cheremisinov, D. Schematic netlist converter / D. Cheremisinov // Proceedings of the Fourth Int. Conf. on Computer-Aided Design of Discrete Devices (CAD DD'2001), Minsk, Nov. 14–16, 2001. – Minsk, 2001. – Vol. 1 – P. 121–125.

10. Кириенко, Н.А. Компиляция функционально-структурных описаний логических схем / Н.А. Кириенко // Автоматизация проектирования дискретных систем: сб. науч. ст. – Минск: Ин-т техн. кибернетики АН Беларуси, 1995. – С. 92–101.

11. Торопов, Н.Р. Преобразование многоярусной комбинационной сети в двухъярусную / Н.Р. Торопов // Логическое проектирование: сб. науч. ст. – Минск: Ин-т техн. кибернетики АН Беларуси, 2000. – Вып. 5. – С. 4–14.

12. Черемисинова, Л.Д. К факторизационному методу синтеза многоуровневых схем на БИС / Л.Д. Черемисинова // Цифровая обработка информации и управление в чрезвычайных ситуациях: материалы Первой Междунар. конф. «Computer-Aided Design of Discrete Devices» (CAD DD'01), Минск, 14–17 ноября 1998 г. – Минск, 1998. – Т. 2. – С. 227–232.

13. Торопов, Н.Р. Минимизация систем булевых функций в классе ДНФ / Н.Р. Торопов // Логическое проектирование: сб. науч. ст. – Минск: Ин-т техн. кибернетики АН Беларуси, 1999. – Вып. 4. – С. 4–19.

14. Кардаш, С.Н. Синтез комбинационных схем из библиотечных элементов / С.Н. Кардаш // Методы логического проектирования: сб. науч. тр. – Минск: ОИПИ НАН Беларуси, 2003. – Вып. 2. – С. 33–40.

15. Кардаш, С.Н. Синтез комбинационных схем, реализуемых в составе базовых матричных кристаллов / С.Н. Кардаш // Материалы Пятой Междунар. конф. «Computer-Aided Design of Discrete Devices» (CAD DD'2004), Минск, 15–17 ноября 2004 г. – Минск: ОИПИ НАН Беларуси, 2004. – Т. 2. – С. 171–178.

16. Goldberg, E. BerkMin: A fast and robust SAT-solver / E. Goldberg, Ya. Novikov // Proceedings of Design, Automation and Test in Europe Conference. – Paris, 2002. – С. 142–149.

17. Романов, В.И. Использование механизма продукции для организации динамического меню в САПР функционально-логических схем / В.И. Романов // Актуальные проблемы радиоэлектроники: научные исследования, подготовка кадров: материалы конференции. – Минск: МГВРК, 2006. – С. 295–297.

Поступила 03.05.06

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
email: bibilo@newman.bas-net.by*

P.N. Bibilo, S.N. Kardash, V.I. Romanov

SyVer – THE SYSTEM FOR MULTI-OUTPUT COMBINATIONAL CIRCUIT SYNTHESIS AND VERIFICATION

A system for synthesis and verification of multi-output combinational circuits in the basis of design libraries of gate arrays is described. The input data of the system are matrix and bracket descriptions of systems of Boolean functions, and VHDL descriptions as well. The SyVer system is attached to LeonardoSpectrum synthesis system and in addition has the programs of logical minimization and verification. The mutual application of SyVer and LeonardoSpectrum allows to obtain logical circuits that have less complexity and delay.