

Раздел VI. Вычислительные комплексы нового поколения и нейрокомпьютеры

УДК 004.4.4

Ю.В. Чернухин, М.Ю. Поленов, Д.В. Булгаков

ИСПОЛЬЗОВАНИЕ МНОГОЯЗЫКОВОЙ ТРАНСЛЯЦИИ ПРИ RTL-СИНТЕЗЕ СБИС НА ЯЗЫКАХ ОПИСАНИЯ АППАРАТУРЫ

Описывается подход к трансляции моделей на этапе RTL-синтеза СБИС, реализуемый на основе средств многоязыковой трансляции. Данный подход позволяет синтезировать модифицируемые трансляторы для перевода поведенческих моделей с различных языков описания аппаратуры в RTL-модели, что существенно расширяет функциональные возможности САПР СБИС.

Многоязыковая трансляция; Мультитранслятор; RTL-синтез; языки описания аппаратуры.

Yu.V. Chernukhin, M.Yu. Polenov, D.V. Bulgakov

USAGE OF THE MULTILANGUAGE TRANSLATION IN VLSI RTL-SYNTHESIS ON HARDWARE DESCRIPTION LANGUAGES

The approach to models' translation in the phase of VLSI RTL-synthesis implemented on basis of the Multilanguage translation tools is described. Such approach allows to create upgradable translators for behavioral models conversion from different hardware description languages into RTL-models and considerably expanding CADs capabilities.

Multilanguage translation; Multitranslator; RTL-synthesis; HDLs.

Введение. Методика проектирования современных цифровых систем на кристалле (СнК) предполагает последовательный синтез моделей системы на нескольких уровнях абстракции при помощи языков описания аппаратуры (ЯОА) [1]. Современные САПР СБИС поддерживают полный цикл синтеза устройств СнК, путем использования собственных синтезаторов кода для генерации на каждом этапе проектирования требуемых описаний соответствующих уровней [2]. Выделяют следующие основные уровни описания: поведенческий, уровень регистровых передач (Register Transfer Level – RTL), уровень списков соединений (компоненты библиотек низкого уровня) и уровень реализации (размещение логических блоков) [3]. При этом необходимо учитывать то, что данные САПР поддерживают только определенный набор ЯОА и их версий, а необходимость добавления нового языка или изменение версии используемых языков требует доработки системы проектирования и синтезаторов кода, что является для пользователей определенной проблемой.

В данной работе рассматривается одно из перспективных направлений решения данной проблемы заключающееся в применении на каждом этапе проектирования СнК средств многоязыковой трансляции (СМТ) [4], разработанных на кафедре Вычислительной техники Таганрогского технологического института ЮФУ. Данные программные средства, названные Мультитранслятором (МТ), отличаются от традиционных новым уровнем абстракции задачи трансляции за счет отделения

описания правил трансляции от технических вопросов ее организации, что предоставляет пользователям возможность самостоятельно создавать и редактировать правила трансляции. При использовании СМТ задача создания трансляторов сводится не к разработке их как отдельных программных продуктов, а к формальному описанию производственных правил трансляции, интерпретируемых в дальнейшем средствами СМТ. Правила трансляции описываются в трансляционных модулях (ТМ), подключаемых к ядру МТ. ТМ Мультитранслятора формируется на языке описания грамматик в виде множества требуемых пользователю грамматических правил исходного языка. Для генерации выходного кода при описании грамматических правил используется язык описания действий Мультитранслятора [4].

Мультитранслятор уже показал свою эффективность при решении задач моделирования [5] и в качестве компилятора компиляторов [6]. Также имеются позитивные результаты исследования возможностей применения МТ для взаимной трансляции описаний моделей на языках одного уровня абстракции на примере языков VHDL и SystemC [7]. В общем случае, Мультитранслятор позволяет генерировать на базе трансляционных модулей исполняемые модули, которые в дальнейшем можно использовать в качестве частных трансляторов для преобразования кодов, написанных на тех или иных ЯОА.

Таким образом, разработка конкретной САПР с поддержкой трансляционных модулей МТ позволяет сделать ее более гибкой по отношению к множеству входных кодов ЯОА, что, в свою очередь позволит ускорить процесс модернизации САПР для новых версий языков и ускорить, таким образом, процедуру проектирования СнК для группы разработчиков, использующих различные ЯОА.

Постановка задачи. В данной работе анализируются возможности использования Мультитранслятора для поддержки одного из основных этапов проектирования систем на кристалле, а именно, RTL-синтеза [2]. Для реализации данного подхода необходимо разработать трансляционный модуль Мультитранслятора и создать в среде МТ частный транслятор, поддерживающий входной набор RTL-синтезируемых конструкций языка описания аппаратуры (в качестве ЯОА был выбран VHDL) и генерирующий описание модели на уровне регистровых пересылок. Синтез уровня регистровых передач представляет собой процесс трансляции программы алгоритмического уровня, заданной набором конструкций ЯОА, в набор соответствующих укрупненным примитивов из выбранного базиса [1]. Совокупность примитивов описывает функциональную схему, которая может быть в дальнейшем использована другими средствами для последующих этапов проектирования СнК (для логического и физического синтеза) и реализована в СБИС [8]. Примитивы RTL-представления должны быть описаны на специальном RTL-языке. В качестве такого языка может быть использован один из форматов обмена проектами электронных схем, например, EDIF.

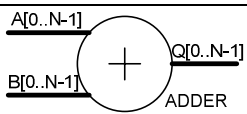
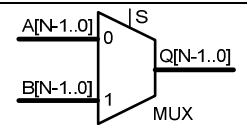
В качестве языка, генерируемого трансляционным модулем МТ, был предложен промежуточный язык, пример конструкций которого, с соответствующими им функциональными элементами, приведен в табл. 1. Каждый оператор такого языка описывает один функциональный элемент схемы, а в качестве параметров оператора выступают имена соединений указанной разрядности, которые объединяют примитивы в схему. Для исследования возможностей среды МТ в качестве входного набора для трансляционного модуля были выбраны следующие синтезируемые конструкции языка VHDL: выражения, операторы условия, присваивания, цикла и другие [9]. В дальнейшем, данный ТМ может быть легко расширен путем добавления новых грамматических правил.

Традиционные синтезаторы кода используют для внутреннего представления данных специальные древовидные графовые структуры, формируемые в ре-

зультате обработки грамматического дерева, полученного в процессе синтаксического разбора [2]. Дальнейшая оптимизация и генерация выходного кода производятся на основании данных внутреннего дерева.

Таблица 1

Примеры конструкций RTL-языка и соответствующих им примитивов

Примитив	Текстовое RTL-обозначение	Описание
	<pre>ADD(in A:<Wire>, in B:<Wire>, out Q:<Wire>)</pre>	N-разрядный знаковый сумматор
	<pre>MUX(in A:<Wire>, in B:<Wire>, in S:<Wire>, out Q:<Wire>)</pre>	Двухвходовый N-разрядный мультиплексор

Предлагаемый подход на основе использования МТ заключается в том, что набор конструкций входного ЯОА разделяется на две группы:

- ◆ конструкции, непосредственно транслирующиеся в выходной RTL-код в процессе построения синтаксического дерева (описания портов, выражения, циклы, операторы условия);
- ◆ конструкции, на основании которых в процессе синтаксического разбора формируются и накапливаются дополнительные массивы данных. Эти данные обрабатываются в те моменты синтаксического разбора, когда их достаточно для оптимизации и генерации фрагментов выходного кода.

Данное обстоятельство позволяет избежать этапа построения полного внутреннего графа данных, присущего многим синтезаторам [2]. Это, в свою очередь, ускоряет процесс синтеза за счет реализации в МТ трансляции конструкций «на лету» и уменьшает объем используемых МТ ресурсов.

Разработка трансляционного модуля Мультитранслятора для осуществления RTL-синтеза. В рамках предложенного подхода к использованию Мультитранслятора при RTL-синтезе СнК, в среде МТ был создан требуемый трансляционный модуль с языка VHDL стандарта IEEE 1076 [10]. Реализованный в МТ подход позволяет посредством стека передавать между грамматическими правилами параметры соединений, которыми связаны элементы функциональной схемы, соответствующие конструкции VHDL.

Далее рассмотрим особенности мультитрансляции отдельных конструкций языка VHDL.

Выражения на VHDL транслируются в комбинационную схему в процессе разбора грамматического дерева. На рис. 1 показан фрагмент дерева разбора простого логического побитового умножения сигналов *A* and *B*. На рис. 1 в прямоугольных блоках описаны действия, производимые МТ при разборе соответствующих символов грамматического дерева. Широкими стрелками изображен процесс передачи параметров между узлами дерева посредством стека. Поскольку терминальным символам сигналов *A* и *B* должны соответствовать некоторые элементы памяти схемы, то при разборе этих символов генерируются соединения *Wire_A* и *Wire_B*, параметры которых передаются вышележащему символу дерева. В этом блоке производится генерация необходимого RTL-оператора и выходного соединения результата *Wire_0*, параметры которого передаются вышележащему символу. Таким образом, в результате обработки рассмотренного дерева выражения, в вершине стека будут храниться параметры соединения, являющегося выходом комбинационной схемы.

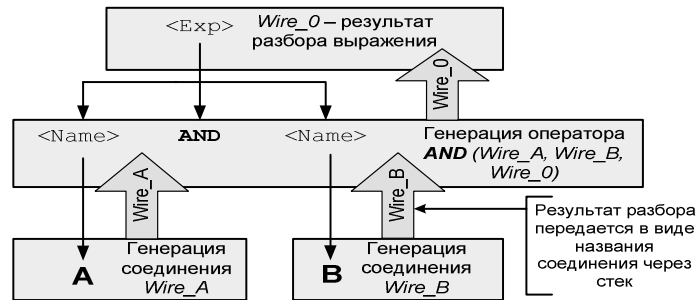


Рис. 1. Дерево разбора выражения A and B

Это соединение в дальнейшем может быть использовано для генерации других RTL-операторов. Необходимо отметить, что среда мультитрансляции позволяет производить упрощение константных выражений «на лету», т.е. в процессе их разбора, передавая через стек не параметры соединения, а вычисленные на нижних уровнях дерева числовые значения подвыражений.

Синтаксическое правило оператора условия IF состоит из трех нетерминальных символов: условия, блока операторов истинности, выполняемых при истинности условия и блока операторов ложности, выполняемых при его ложности [8,10]. Условие оператора является выражением, поэтому разбор его осуществляется по схеме разбора выражений, рассмотренной выше. После разбора условия в стеке хранятся параметры соединения результата условия. В процессе разбора символа оператора IF, всем символам операторов, находящихся в блоке истинных операторов, передаются параметры соединения результата условия. При разборе этих операторов условия комбинируются с условиями нижележащих операторов IF или подаются на вход разрешения записи для операторов элементов памяти. Операторам блока ложности передаются соединения с выхода RTL-примитива NOT (инвертирующего условия), код которого генерируется также в процессе разбора символа оператора IF.

Оператору присваивания языка VHDL соответствует некоторый элемент памяти, например регистр, которому ставится в соответствие имя сигнала-приемника оператора присваивания [8]. В роли источника оператора присваивания выступает некоторое выражение, транслируемое МТ в комбинационную схему. Результат выражения подается на вход данных памяти. Важно отметить, что трансляция операторов присваивания в ТМ осуществляется в два этапа.

На первом этапе выполняется разбор символа сигнала-источника оператора присваивания и выражения-приемника с получением соединения результата выражения. Далее эти данные совместно с параметрами соединений условия и синхросигнала, полученными от вышележащих в грамматическом дереве символов условия, добавляются в локальную таблицу пересылок.

Необходимость формирования такой таблицы обусловлена тем, что в теле процесса могут существовать несколько операторов присваивания с одинаковыми приемниками сигнала или приемниками, имеющими общие разряды. В этом случае необходима дополнительная обработка пересылок и формирование дополнительной логики приоритетов присваивания, которые выполняются на втором этапе трансляции операторов присваивания. Эти действия осуществляются после завершения разбора символа процесса. Также на этом этапе формируются выходные RTL-операторы элементов памяти. Тип элемента памяти определяется по алгоритму, изображенному на рис. 2.

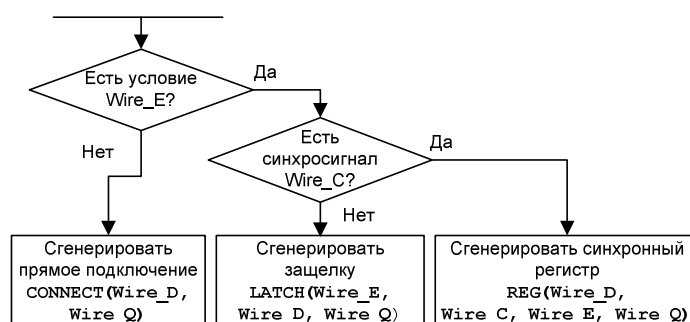


Рис. 2. Алгоритм определения типа элемента памяти

Трансляция операторов цикла [2] в трансляционном модуле МТ осуществляется разверткой, при этом каждый оператор внутри тела цикла преобразуется в набор параллельных RTL-блоков, количество которых равно количеству итераций цикла. Данный подход предполагает, что количество итераций цикла и все значения счетчика цикла должны быть известны до начала разбора символов операторов, находящихся внутри тела оператора цикла.

Метод развертки был реализован с помощью таблиц значений локальных переменных для каждого уровня вложенности цикла. Каждый счетчик цикла для каждого уровня вложенности цикла имеет свою таблицу, хранящую все возможные значения этой переменной. Все выражения и операторы, использующие эту переменную на текущем уровне вложенности цикла, обязаны повторять генерацию RTL-кода для каждого значения такой переменной. Генерация новой таблицы локальных переменных производится в процессе разбора символа оператора цикла и использует таблицы локальных переменных текущего уровня вложенности.

В разработанном трансляционном модуле Мультитранслятора также были реализованы и другие языковые конструкции VHDL. Объем кода модуля VHDL-RTL составил 3467 строк описания. Тестирование разработанного ТМ было проведено при трансляции набора VHDL-моделей цифровых устройств в RTL-описание. Результаты трансляции моделей сравнивались со схемами RTL-уровня, полученными для этих устройств в среде синтеза Altera Quartus II [11].

На рис. 3 приведены результаты трансляции трансляционным модулем фрагмента исходного кода на VHDL и схема RTL-уровня, полученная для этого фрагмента в среде Quartus II. Сравнение полученных результатов позволяет утверждать, что функциональность синтезируемых моделей при использовании мультитрансляционного подхода сохраняется.

Заключение. Замена стандартных синтезаторов кода частными трансляторами, созданными на основе ТМ в среде Мультитранслятор, позволит сделать САПР СБИС более гибкими и расширяемыми. В этом случае, при возникновении у разработчика необходимости изменить входной или выходной язык описания СнК, следует только модифицировать код ТМ и подключить его к ядру МТ, в отличие от стандартного подхода, который требует модификации всей среды САПР и создания отдельного транслятора для выбранной пары языков.

Частные трансляторы, созданные в среде МТ, также могут быть использованы для быстрого переноса проектов из одних САПР СБИС в другие. Это может быть достигнуто разработкой трансляционных модулей, генерирующих на выходе описания в общепринятых форматах обмена проектами электронных схем (например, EDIF, CIF, DXF). Создание таких модулей для одного входного ЯОА и набора выходных языков существенно упрощается, поскольку разработчику не

требуется писать отдельный транслятор для каждой пары языков. Достаточно взять трансляционный модуль для одной пары и изменить операторы генерации выходного кода для конкретного формата, таким образом, получив новый ТМ.

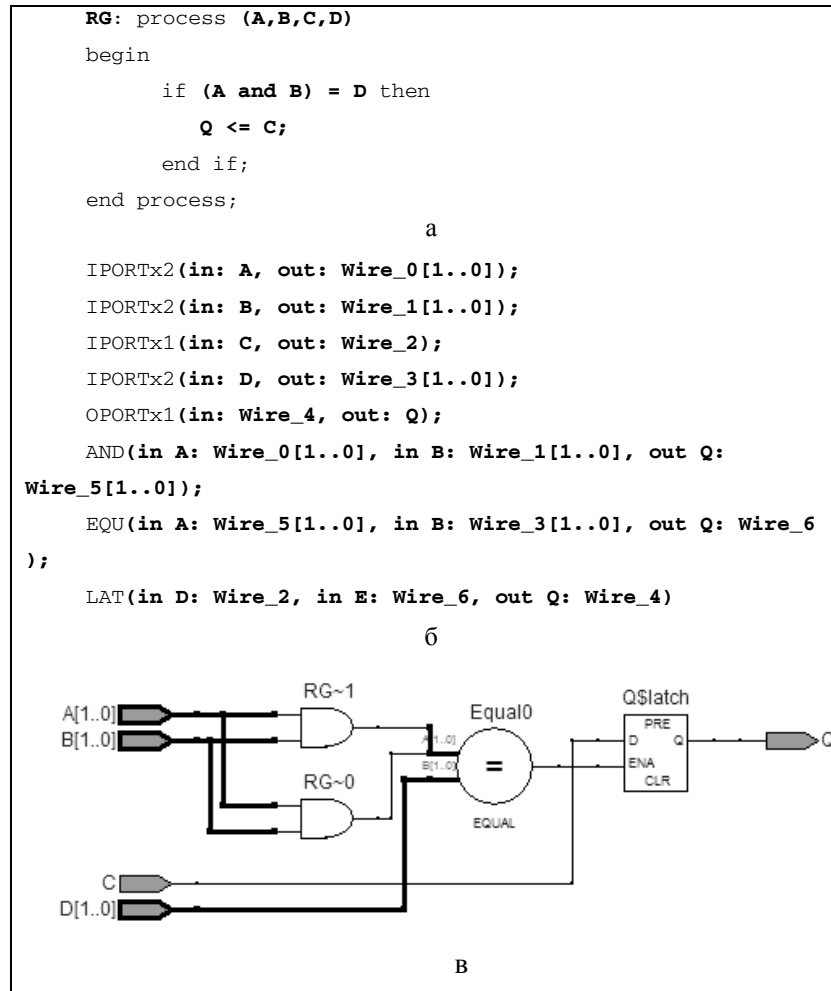


Рис. 3. Пример использования трансляционного модуля МТ для RTL-синтеза:
 а – фрагмент исходного VHDL-кода поведенческой модели;
 б – результат работы частного синтезатора кода, созданного в среде МТ;
 в – RTL-схема, полученная с помощью Altera Quartus II.

Применение МТ возможно также и на других промежуточных этапах проектирования систем на кристалле, например для высокоуровневого синтеза [1]. Замена большинства компонент САПР вычислительных устройств на частные трансляторы, синтезируемые в среде МТ, позволит сделать средство проектирования СнК более гибким и модернизируемым. А подходы, выработанные при создании синтезаторов кода с использованием МТ, могут быть в дальнейшем использованы при разработке полноценного средства RTL-синтеза.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Максфилд К.* Проектирование на ПЛИС. Архитектура, средства и методы. Курс молодого бойца. – М: Издательский дом «Додэка-XXI», 2007. – 408 с.
2. *Sharp R.* Higher-Level Hardware Synthesis. – Berlin: Springer, 2004. – 198 p.
3. *Томас Ф., Иванов А.* САПР микроэлектроники. Этапы большого пути // Электроника. Наука, Технология, Бизнес. – М: РИЦ «Техносфера», 2006. – № 3. – С. 82-85.
4. *Чернухин Ю.В., Поленов М.Ю.* Инструментальная подсистема многоязыковой трансляции виртуальных моделирующих систем // Известия ТРТУ. – 2004. – № 3 (38). – С. 115-120.
5. *Чернухин Ю.В., Гузик В.Ф., Поленов М.Ю.* Подход к формированию внешних библиотек сред виртуального моделирования на базе мультязыковой трансляции // Вестник компьютерных и информационных технологий. – 2008. – № 10. – С. 2-12.
6. *Чернухин Ю.В., Поленов М.Ю., Левченко Д.В.* Возможности применения среды мультитрансляции в качестве компилятора компиляторов // Искусственный интеллект. – Донецк: Изд-во «Наука і освіта». 2008. – № 4. – С. 712-720.
7. *Чернухин Ю.В., Поленов М.Ю., Булгаков Д.В.* Использование мультязыковой трансляции при конверсии моделей, представленных на языках описания аппаратуры // Искусственный интеллект. – Донецк: Изд-во «Наука і освіта». 2009. – № 4. – С. 462-471.
8. *VHDL Synthesis Reference.* Altium Ltd, 2005. [Электронный ресурс]. Режим доступа: <http://www.altium.com/files/learningguides/TR15%20VHDL%20Synthesis%20Reference.pdf>
9. *Сергиенко А.М.* VHDL для проектирования вычислительных устройств. – М: DiaSoft, 2003. – 208 с.
10. *IEEE1076. Синтаксис языка VHDL.* VHDL-обучающий портал. [Электронный ресурс]. Режим доступа: <http://www.bsuir.by/vhdl/syntax/>.
11. *Altera Quartus II.* Altera Corporation, 2010. [Электронный ресурс]. Режим доступа: <http://www.altera.com/products/software/sfw-index.jsp>.

Чернухин Юрий Викторович

Технологический институт Федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге
E-mail: chernukhin@dce.tsure.ru
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 88634371656.

Поленов Максим Юрьевич

E-mail: polenov@dce.tsure.ru

Булгаков Данил Вячеславович

E-mail: buda@mail.ru

Chernukhin Yuri Victorovich

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”
E-mail: chernukhin@dce.tsure.ru
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: +78634371656.

Polenov Maxim Yuryevich

E-mail: polenov@dce.tsure.ru

Bulgakov Danil Vyacheslavovich

E-mail: buda@mail.ru