

ИЗВЛЕЧЕНИЕ ГРАФА СОЕДИНЕНИЙ КОМПОНЕНТ ИЗ ОПИСАНИЯ СБИС НА ЯЗЫКЕ VHDL

Черемисинова Л. Д., Черемисинов Д. И.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: {cld, cher}@newman.bas-net.by

Рассматривается задача, возникающая при интеграции программ САПР от разных поставщиков в среду проектирования СБИС. При интеграции в среду проектирования существенным ограничением является недоступность внутренних функциональных возможностей программы, так что взаимодействие программ в процесс проектирования состоит только в обмене данными. Программы САПР взаимодействуют, обмениваясь сообщениями, представляющими собой описание проектируемого устройства на языке проектирования аппаратуры или в специальном обменном формате. Интерфейсом для объединения инструментов САПР может служить язык обмена информацией, в качестве которого часто используется язык описания аппаратуры VHDL. Предлагается метод и программа, которые извлекают граф соединенный компонент из описания СБИС на VHDL. Граф соединенный может быть использован для построения структурного описания в формате, требуемом сопрягаемой программой САПР.

ВВЕДЕНИЕ

Наиболее известными и широко используемыми языками описания оборудования (Hardware description language - HDL) систем автоматизации проектирования (САПР) являются языки Verilog и VHDL [1]. На этих языках можно описать как первоначальный проект СБИС, так и его промежуточные представления в процессе подготовки СБИС к изготовлению. Поскольку на HDL можно описывать как структуру, так и функции устройства, HDL часто используются для представлений промежуточных результатов проектирования, формируемых программами САПР СБИС. Для больших проектов неизбежно возникают проблемы, связанные с комплексированием инструментов САПР, которые решаются путем создания специальных программ сопряжения. В задачу этих программ входит анализ и изменение описания СБИС на HDL.

Анализ VHDL – это весьма сложная задача из-за синтаксической сложности этого языка [3]. Существуют несколько коммерческих анализаторов VHDL, которые позволяют исследователям, проектировщикам и другим пользователям строить инструменты проектирования и использовать VHDL-описания как обменный формат. Интерфейсы этих анализаторов отличаются по формату данных для представления результатов анализа и методам доступа к этим данным. Семантически текст на языке VHDL описывает некоторое цифровое устройство. В процессе проектирования используются несколько описаний, отличающихся уровнем абстракции поведения устройства. Фактически описания разного уровня абстракции отличаются языковой конструкцией, называемой в VHDL архитектурой. Каждый уровень требует особых языковых средств задания архитектуры, составляющих некоторое подмножество исходного языка описания проектирования аппаратуры.

I. СТИЛИ КОДИРОВАНИЯ АРХИТЕКТУРЫ УСТРОЙСТВА НА VHDL

Языковые средства VHDL для задания архитектуры определенного уровня абстракции, составляют некоторое подмножество VHDL. Эти подмножества называются стилями кодирования. Основных стилей кодирования – три: 1) поток данных, 2) поведенческий. 3) структурный. Подмножества языка VHDL не составляют особых языков и заданы как рекомендации для использования проектировщиками с определенной целью и для фиксации результатов этапов проектирования. Для архитектуры структурного стиля характерно использование операторов создания экземпляров компонентов (port map). Стиль потока данных (data flow) характеризуется тем, что используются оператор назначения сигналов, логические и арифметические операторы. Стиль для описания объекта проектирования в алгоритмической форме путем задания преобразования входных данных в выходные, называется поведенческим. Элементы языка VHDL, которые могли бы быть синтезированы средствами синтеза (компиляторами) на уровне регистровых передач (register transfer level) составляют синтезируемое подмножество VHDL. В описании на синтезируемом подмножестве могут смешиваться все три стиля кодирования архитектуры.

II. ЗАДАЧА СИНТАКСИЧЕСКОГО АНАЛИЗА СИНТЕЗИРУЕМОГО ПОДМНОЖЕСТВА VHDL

В теории формальных языков синтаксический анализ (разбор, парсинг – parsing) – это процесс определения того, может ли строка символов быть получена из множества продукций (set of production rules), называемых грамматикой. Однако, часто задачей синтаксического анализа заданного текста (последовательности символов) считается построение дерева вывода. Дерево вывода описывает синтаксическое стро-

ение текста в соответствии с правилами заданной грамматики. Для языка VHDL источником грамматики может быть стандарт языка – LRM (Language Reference Manual), задающий синтаксис и семантику языка. Граф соединений компонент может быть построен в результате обхода части дерева вывода, соответствующей архитектуре структурного (или смешанного) стиля.

Выводом (derivation) называется последовательность шагов, начинающаяся с начального символа грамматики, и заканчивающаяся строкой из терминальных символов. Каждый шаг состоит в применении правила грамматики, т.е. замене в исходной строке символа нетерминала правой частью правила. Строки, получаемые в результате применения правил грамматики, называются сентенциальными формами. Вид дерева вывода для заданных текста и грамматики зависит от метода синтаксического анализа – стратегии разбора. Имеются два класса стратегий: 1) левосторонняя (LL): для применения правила всегда выбирается самый левый нетерминал сентенциальной формы, 2) правосторонняя (LR): для применения правила всегда выбирается самый правый нетерминал сентенциальной формы. Синтаксический анализ языка VHDL все еще не является решенной проблемой на практике, несмотря на изощренность современных стратегий синтаксического анализа и долгую историю академических исследований.

Для детерминированности результата грамматики должна удовлетворять правилам избранной стратегии синтаксического анализа. Тип синтаксического анализатора со стратегией снизу вверх (LR парсер) распознает любую строку контекстно-свободного языка за линейное время. Для применения этой стратегии нужна леворекурсивная грамматика. Нисходящий анализ (LL парсер) – это стратегия определения принадлежности входной строки к некоторому формальному языку, в которой правила формальной грамматики применяются, начиная с начального символа грамматики, пытаюсь корректно разбить исходную строку на непосредственные составляющие. Для применения этой стратегии нужна праворекурсивная контекстно-свободная грамматика. Нисходящий рекурсивный парсер неприменим для разбора леворекурсивных грамматик. Праворекурсивную грамматику в нисходящей стратегии можно применить, если читать исходный текст в обратном порядке.

III. АНАЛИЗАТОР VHDL, ГЕНЕРИРУЮЩИЙ ГРАФ СОЕДИНЕНИЙ КОМПОНЕНТ

Известны методы и программы, которые по заданной контекстно-свободной грамматике могут построить программу, генерирующую дерево вывода для заданного текста, удовлетворяющего грамматике. Эти программы требуют определенной формы грамматики. Автоматическое созда-

ние синтаксического анализатора, выполняющего анализ за линейное время, возможно только для подмножества контекстно свободных грамматик. Почти для любой произвольной грамматики можно найти немного другую грамматику, которая генерирует тот же язык и допускает синтаксический анализ за линейное время. Однако поиск такой грамматики почти всегда требует вмешательства человека и не может быть автоматизирован.

Генераторы анализаторов могут построить более компактную форму результатов анализа, вместо дерева синтаксического анализа для пост-синтаксической обработки, если модифицировать грамматику добавлением действий с побочным эффектом, создающих в памяти представление результатов в требуемой форме – в виде графа соединений компонент. Эта модификация грамматики требует вмешательства человека и тоже не может быть автоматизирована.

Наиболее известный генератор синтаксических анализаторов по заданной грамматике называется компилятором компиляторов yacc - bison. Этот компилятор компиляторов использует стратегию синтаксического анализа, называемую LALR(1). Аббревиатура означает синтаксический анализ «LR с просмотром на 1 символ вперед». Грамматика из LRM VHDL имеет форму, не подходящую для стратегии LALR(1) [2]. Так как интерфейсом для объединения инструментов САПР служит синтезируемое подмножество языка VHDL, можно исследовать грамматику для распознавания этого подмножества. Большинство приведенных в литературе грамматик леворекурсивны и требуют генераторов, использующих стратегию LL.

В результате опробования нескольких грамматик из литературы, подходящим для компилятора yacc - bison оказалась грамматика из открытой САПР СБИС Alliance [3]. Эта грамматика была дополнена действиями, позволяющими создать в памяти представление результатов анализа в виде графа соединений компонент.

IV. ЗАКЛЮЧЕНИЕ

Разработанная программа анализа синтезируемого подмножества VHDL была использована в качестве компонента конвертера, преобразующего описание СБИС в форму на языках VHDL, EDIF, SF.

V. СПИСОК ЛИТЕРАТУРЫ

1. VHDL'93. IEEE Standard VHDL Language Reference Manual. -IEEE Std 1076-1993, 2002. - 289 p.
2. Lorenc, Lubos et al. A Note on the Parsing of Complete VHDL-2002 // Workshop on Formal Models (2007) [Электронный ресурс] / <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-255/paper11.pdf>
3. Alliance – A Free VLSI/CAD System [Электронный ресурс] / <http://www-soc.lip6.fr/equipement/logiciels/alliance>.