

Перспективные направления решения задачи синтеза HDL-программ в САПР РЭА

*Григорьев А.В., к.т.н., доц., Грищенко Д.А., асс.,
Донецкий национальный технический университет,
grigorie@pmi.dgtu.donetsk.ua, kosheleva_daryav@mail.ru*

Рассматриваются стандартные существующие подходы синтеза vhdл-файлов в САПР. Приведен обзор существующих интеллектуальных подходов решения данной задачи и определены их недостатки. Определено дальнейшее направление исследования.

Введение

Тенденции развития систем автоматизации проектирования средств вычислительной техники привели к возникновению целого класса языков, позволяющих проектировать алгоритмы требуемых вычислительных устройств с последующим синтезом на их основе топологии ПЛИС. Язык HDL является одним из наиболее популярных языков такого класса. В настоящее время в САПР используется версия языка HDL, ориентированная на возможность графического отображения структурной компоненты программ – язык VHDL. Такой язык позволяет совместить достоинства графических моделей устройств и текстовых описаний алгоритмов. Типичные проектные процедуры в современных САПР радиоэлектронной аппаратуры (РЭА), касающиеся HDL-программ, это: ручное построение программы или ее автоматический синтез, корректировка и верификация алгоритмов функционирования (имитационное моделирование) и документирование. Инструментальные средства процедур ручного построения, корректировки и документирования HDL-программ являются наиболее развитой частью существующих САПР и обеспечивают полноценное и эффективное выполнение требуемых функций. Как пример можно упомянуть наиболее развитую САПР такого класса Active HDL. Имеется ряд работ, посвященных модификации VHDL как языка с целью внесения в него модельных временных задержек для обеспечения асинхронного имитационного моделирования. Однако, наиболее трудоемкой и наименее развитой является процедура автоматического синтеза VHDL-программ. Наиболее эффективным подходом для решения данной задачи является применение методов

искусственного интеллекта. В настоящее время имеются некоторые интеллектуальные методы, обеспечивающие синтез таких программ. Однако данная задача еще далека от полного решения. Т.о., актуальной является задача построения интеллектуальных средств и методов, позволяющих более эффективно решать задачу синтеза VHDL-программ. Цель работы: 1) Анализ методов автоматизации проектирования в САПР Active HDL; 2) Анализ альтернативных подходов синтеза HDL-файлов других САПР; 3) Анализ интеллектуальных средств и методов автоматизации синтеза HDL-программ; 4) Определение перспективных направлений решения задачи синтеза HDL-программ.

1. Анализ методов автоматизации проектирования в САПР Active HDL

Среди имеющихся средств автоматизации создания моделей на языке VHDL в различных САПР можно назвать [1]: использование шаблонов, макросов, мастеров для синтеза текстов программ; формирование языковых описаний по графической модели алгоритма, представленной в виде блок-схемы; использование наборов подключаемых библиотек программ. Следует отметить, что из названных средств мастера являются самым развитым методом синтеза решений в САПР на базе VHDL. Однако, разработка каждого нового мастера, при условии его полноценного построения, есть задача разработки маленького специализированного САПР, ориентированного на некоторый класс решений (устройств). Т.о., такой процесс передачи знаний имеет высокую трудоемкость. С целью повышения эффективности процессов создания мастеров можно применить технологию создания мастера как экспертной системы, способной обучаться, т.е. перенимать опыт проектировщика на ряде примеров или приобретать знания напрямую от эксперта.

2. Анализ альтернативных подходов синтеза HDL-файлов в других САПР

Одним из типичных подходов к автоматизации формирования HDL-программ является синтез HDL-файлов на основе граф-схемы алгоритмов (ГСА), представляющих собой автоматную модель вычислительного устройства. [2]. Данный подход может быть классифицирован как вариант специализированных CASE – технологий, обеспечивающих синтез программного обеспечения на требуемом языке программирования на основе диаграмм, представляющих алгоритм работы программы.

Еще одним типичным подходом является синтез VHDL-кода на основе диаграмм состояний UML [3]. Недостатком такого рода подходов является практически ручное создание диаграмм, описывающих алгоритм работы вычислительного устройства, а также необходимость знания особенностей представления необходимой модели (типов блоков диаграмм и отношений между ними) в инструментальной среде разработки. В качестве аналогичного подхода можно рассматривать синтез VHDL-файлов на основе программ, написанных на языках высокого уровня: C, C++, решающих дополнительно оптимизационные задачи, например, позволяющие выполнять распараллеливания алгоритмов функционирования устройства [4]. Еще одним эффективным подходом в построении VHDL-файлов есть механизм перевода принципиальных схем, подготовленных в САПР РЭА типа ORCAD. В этом случае такая САПР имеет команды меню, позволяющие сгенерировать соответствующую VHDL-программу по построенной ранее принципиальной схеме. Наиболее обще процесс синтеза структуры цифрового управляющего устройства на основе содержательного перечня микроопераций и отличающийся более эффективными графическими моделями представления алгоритма функционирования устройства, представлен в [5]. Общими недостатками существующих подходов являются:

- отсутствие средств автоматизации синтеза нужных VHDL- программ, реализующих требуемый алгоритм работы целевого вычислительного устройства по ряду параметров, определяющих его структурные и функциональные особенности, т.е. построение решения по требуемому техническому заданию;

- необходимость достаточно высокой подготовки и уровня знаний специфики инструментальных средств построения графических диаграмм, описывающихся требуемый алгоритм работы устройства.

- полученные решения фактически есть воплощения неформальной, эвристической методики проектирования, которой пользуется проектировщик.

С другой стороны, следует отметить, что в настоящее время имеется тенденция создания библиотек VHDL-программ, реализующих те или иные аппаратные решения. Данные решения, как правило, апробированы на практике, т.е. имеют высокий уровень достоверности. Посредством библиотек решения могут накапливаться и систематизироваться для облегчения доступа. Данные библиотеки позволяют многократно использовать удачные решения или модифицировать их с учетом новых задач. Они являются объектами интеллектуальной собственности, т.е. могут продаваться и покупаться. Эти библиотеки доступны как в различных САПР или на web-порталах. Можно сказать, что данные биб-

лиотеки являются аккумулятором успешного опыта проектирования тех или иных устройств по тем или иным эвристическим методикам проектирования. Но выбор требуемого решения из библиотек, как правило, производится вручную. Таким образом, актуальной является задача автоматического выявления и формализации эвристических методик проектирования в форме баз знаний, предназначенных для автоматического проектирования требуемого класса устройств. Следует отметить, что синтаксические и семантические особенности VHDL – программ, попадающих в данные библиотеки, напрямую зависят от источника, из которого данные программы попадают в библиотеку. Подводя итог выше сказанному, можно назвать следующее источники:

- программы, автоматически синтезируемые САПР типа ORCAD, в которых явно используются наименования корпусов микросхем (например, процессоров), наименования резисторов, транзисторов и прочих компонент схем, выбранных из соответствующих справочников; как правило, такие модели являются одноуровневыми, т.е. имеют вид одной большой принципиальной схемы;

- программы, созданные с помощью САПР VHDL, где пользователь, как всякий программист, может использовать произвольные идентификаторы для переменных и процедур, не имеющих ни какого соответствия с известными типичными решениями; стандартные элементы могут иметь вид базовых функций VHDL, типа «и-не», «или-не» или являться стандартными именами из подключаемых библиотек, например, регистров, счетчиков и т.д.;

- программы, созданные при помощи синтеза по ГСА или подобными инструментальными средствами; в этом случае также имеет место возможность вводить произвольную идентификацию блоков, входов-выходов, но - с использованием типичных базовых элементов.

Таким образом, можно сказать, что в данных VHDL-программах имеются как типичные библиотечные идентификаторы с известной семантикой, так и - многочисленные идентификаторы, семантика которых привязана к конкретной разработке и не имеет универсального характера. Это означает, что если, например, имеются две VHDL – программы, описывающие структуру одного и того же устройства в разных контекстах применения, или - описывающих близкие структуры, не имеющих принципиальных отличий и относящиеся к одному типу, то они могут быть представлены с использованием как различных идентификаторов и с различным порядком описания общих структур.

В качестве вывода можно сказать, что имеет место следующая актуальная задача:

- необходим инструментальный комплекс, способный обеспечить извлечение методик проектирования из существующих библиотек решений VHDL-программ;

- комплекс должен быть способен обеспечить достаточно малую трудоемкость процесса извлечения методик проектирования, т.е. быть адаптирован на «глупого» эксперта [6];

- важно учитывать уровень способности проектировщика помочь системе в данном процессе, учитывая при этом достаточно мелкие градации его квалификации в этом вопросе, что позволит эффективно адаптировать инструментальный комплекс на специфику условий извлечения методики.

3. Анализ интеллектуальных средств и методов автоматизации синтеза VHDL-программ

Рассмотрим существующие средства и методы искусственного интеллекта, способные обеспечить решение поставленной выше задачи. Фактически необходимо решать два класса задач: 1) адаптация на уровень квалификации пользователя в пределах возможности «глупого» эксперта, что фактически есть задача построение модельной процедуры синтеза различного уровня полноты, исходя из возможностей эксперта в предметной области; фактически речь идет о полноте множества технических заданий (ПОС), множества решений-прототипов (ЦПС) и механизме реализуемости, т.е. взаимосвязи между ПОС и ЦПС; 2) адаптация на источник получения программ фактически есть задача извлечения знаний из текстов, имеющих различные доли формальных (однозначных) и неформальных (субъективных, неоднозначных) компонент. Можно выделить два класса инструментальных средств и методов, позволяющих решить задачу номер два в данной специфической постановке: 1) формальные методы теории синтаксического анализа и грамматик для формальных языков; 2) методы и подходы извлечения знаний из естественно языковых текстов [7]. Достоинствами формальных методов являются многочисленные механизмы лексического, синтаксического анализа текстов, а недостатками - отсутствие средств семантического анализа. По поводу методов извлечения знаний из языковых текстов можно сделать вывод, что формируемые модели представления знаний, извлеченные из текста, не вполне соответствуют специфике VHDL, кроме того, в этом случае решается задача извлечения знаний, но не решается задача обобщения знаний, т.е. - задача построения базы знаний. Достоинством естественно-языковых методов является наличие многочисленных способов формирования моделей объекта как резуль-

тата извлечения знаний. Специфика поставленной выше задачи позволяет сделать вывод, что ни один из двух перечисленных выше классов не обеспечивает решение поставленной задачи. Таким образом, необходим третий подход, обеспечивающий решение задачи с наличием формальных и неформальных компонент. Фактически необходима разработка соответствующего инструментария. Следует отметить наличие такого подхода, представленного в работах [1,8], обеспечивающих в целом решение данной задачи. Сделаем краткую характеристику предлагаемого подхода: 1) использование теоретико-множественных операций над порождающими грамматиками как механизма обобщения, классификации и выявления свойств первого рода, задающих отличия сравниваемых структур (механизм создания ПОС); 2) использование предикативных грамматик как механизма обеспечения ввода множеств продукционных зависимостей над грамматиками, представленными в виде и-или дерева, что соответствует механизму вывода в модуле знаний, отражающего проектную процедуру синтеза. Данный подход обеспечивает: 1) решение поставленной выше задачи в условиях источника VHDL-программ, синтезированных в среде САПР типа ORCAD без какой-либо субъективной неформальной компоненты; 2) автоматическое построение ПОС, состоящее из свойств первого рода, фактически являющимися или-узлами получившегося при обобщении и-или дерева, что соответствует подградации совсем «глупого» эксперта [1]. Недостатки данного подхода: 1) не учитываются более мелкие градации «глупого» эксперта, что не позволяет достичь высокой эффективности процесса адаптации на условия создания интеллектуальной надстройки; 2) не обеспечивается возможность учета различных источников появления VHDL-программ и наличия в них субъективных компонент, связанных с описанием семантически эквивалентных или близких решений, но с наличием разницы в идентификации и порядке описания.

Т.о., можно сделать вывод, что построение методов синтеза VHDL-файлов, лишенных названных недостатков, есть важное перспективное направление.

Заключение

В работе выполнен анализ подходов синтеза VHDL-файлов в существующих САПР. Приведен обзор интеллектуальных подходов решения данной задачи и выделены их недостатки. Как дальнейшее направление исследования можно определить построение методов, лишенных указанных недостатков.

Литература

1. Григорьев А.В., Кошелева Д.А. Интеллектуализация процесса проектирования аппаратуры средствами языка VHDL / Наук. праці ДонНТУ. Серія IKOT. Випуск 93. – Донецьк: 2006. – С. 99-105.
2. Зеленёва И.Я., Дорожко Л.И., Мирошкин А.Н. Система автоматизированного проектирования композиционных микропрограммных устройств управления // Наукові праці Донецького національного технічного університету. Серія: Проблеми моделювання та автоматизації проектування. – Донецьк, 2007. – С. 54-61.
3. D.H.Akehurst, O.Uzenkov, W.G.Howells, K.D.Mcdonald-Maier2, B.Bordbar Compiling UML State Diagrams into VHDL: An Experiment in Using Model Driven Development // Forum on specification and Design Languages, FDL 2007, September 18-20. – Barcelona, Spain, 2007. – P. 219-224.
4. Дубров Д.В., Штейнберг Р. Б. Экспериментальный конвертер с языка C в hdl на основе диалогового высокоуровневого оптимизирующего распараллеливателя // Доклады пятой международной конференции «Параллельные вычисления и задачи управления», Москва РАСО '2010, 26-28 октября. – Москва. – С. 865-870.
5. Зацелкин К.В., Иванова Е.Н. Информационная технология автоматизированного проектирования цифровых управляющих устройств // Електромашинобудування та електрообладнання: Міжвідомчий науково-технічний збірник. – 2009. – Вип. 72. – С. 68-72.
6. Григорьев А.В. Пути создания интеллектуальных САПР при различных уровнях квалификации экспертов /Научно-теоретический журнал «Искусственный интеллект», №3, 2005. – Донецьк: ИПИИ, 2005. – С. 758–763.
7. Алексеев С.С., Морозов В.В., Симаков К.В. Методы машинного обучения в задачах извлечения информации из текстов по эталону // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XI Всероссийской научной конференции RCDL'2009. – Петрозаводск: КарНЦ РАН, 2009. – С. 237-246.
8. Григорьев А.В. Теоретико-множественные операции над грамматиками как механизм работы со знаниями в интеллектуальных САПР. Вісник Східноукраїнського національного університету імені Володимира Даля, N 2(48). Луганск, ВУТУ, 2002. С. 186-194.

