

УДК 004.896

КЛАССИФИКАЦИЯ ИСТОЧНИКОВ ЗНАНИЙ О
МЕТОДИКАХ ПРОЕКТИРОВАНИЯ VHDL-ПРОГРАММ В
ИНСТРУМЕНТАЛЬНОЙ ОБОЛОЧКЕ ПО СОЗДАНИЮ
ИНТЕЛЛЕКТУАЛЬНЫХ САПР В РЕЖИМЕ «ГЛУПОГО»
ЭКСПЕРТА

А.В.Григорьев, Д.А.Грищенко
Кафедра ПМИ, ДонНТУ
grigorie@r5.dgtu.donetsk.ua, darya.grischenko@gmail.com

В статье выполнен анализ существующего уровня автоматизации генерации VHDL-программ, приведена классификация источников знаний (библиотек программ), определены уровни квалификации «глупого» эксперта, предложены оценки эффективности функционирования базы знаний, соответствующей требуемой методике проектирования некоторого типа изделия.

Введение

Как отмечено ранее [1], в настоящее время наблюдается тенденция создания библиотек VHDL-программ, реализующих те или иные аппаратные решения. Данные библиотеки позволяют многократно использовать удачные решения или модифицировать их с учетом новых задач. Эти библиотеки доступны как в различных САПР, так и на web-порталах. Синтаксические и семантические особенности VHDL-программ напрямую зависят от источника, из которого данные программы попадают в библиотеку. Таким образом, возникает необходимость классифицировать существующие источники знаний (библиотеки программ), с целью последующего их использования для создания баз знаний экспертных систем (БЗ ЭС). Кроме того, важно учитывать уровень способности проектировщика помочь инструментальной оболочке в процессе создания БЗ, учитывая при этом градации его квалификации, что позволило бы эффективно адаптировать инструментальный комплекс на специфику условий извлечения методики.

В качестве целей данной работы рассматриваются следующие задачи:

1) выполнить анализ существующего уровня автоматизации генерации VHDL- программ в существующих САПР;

2) классифицировать существующие возможные источники знаний о методиках проектирования VHDL–программ, заданные как набор прецедентов (библиотеки, программы и т.п.), которые могут быть использованы для создания БЗ ЭС, способной генерировать VHDL–код по техническому заданию пользователя;

3) определить уровни квалификации «глупого» эксперта и его возможности в наполнении БЗ;

4) ввести оценки эффективности функционирования БЗ, соответствующей требуемой методике проектирования некоторого типа изделия.

1. Анализ существующего уровня автоматизации генерации VHDL- программ в существующих САПР

Следует отметить, что существуют различные интеллектуальные средства автоматизации формирования VHDL, Verilog, SystemC – модулей. Так, в САПР Active-HDL имеются такие инструменты, как Language Assistant (языковой помощник) и IP Core Generator (генератор интеллектуальных блоков-ядер). Языковой помощник содержит некоторое множество шаблонов типовых конструкций языков HDL, из которых можно создавать исходный код, не вникая в тонкости используемого языка.

Однако, Language Assistant позволяет генерировать только основную структуру шаблона и не дает возможность задавать входные и выходные порты, но есть возможность сделать шаблон диалоговым.

В пакете OrCAD так же имеется языковой помощник — VHDL Samples, который позволяет описывать один входной PORT1_NAME и один выходной PORT2_NAME сигналы. Описания недостающих портов легко добавить при помощи копирования.

В САПР Xilinx ISE тоже имеется такого рода инструмент - Language Templates, но в нем подробно представлены описания портов (входных, выходных и двунаправленных), как для одиночных, так и для шинных сигналов [3].

В качестве недостатка языкового помощника САПР Active-VHDL нужно отметить необходимость знания о месте расположения шаблонов в библиотеке. Кроме того, для описания недостающих в шаблоне портов необходимы знания языка VHDL.

Результатом работы второго инструмента (генератор интеллектуальных блоков-ядер), повышающего эффективность создания HDL-блоков, является полностью готовый для применения модуль.

Такие модули могут использоваться в любой VHDL- или Verilog-системе, они построены на синтезируемом подмножестве языка, а значит, пригодны для инструментов синтеза и ПЛИС-технологий [3].

IP Core генератор осуществляет работу в три этапа:

- 1) проверяется наличие ошибок при вводе имен и значений параметров;
- 2) выполняется собственно этап генерации;
- 3) генератор создает необходимые файлы и папки, куда и помещает результат своей работы.

В результате генерации получается три файла:

- 1) файл с расширением *.vhd (содержит сгенерированный VHDL-код);
- 2) файл с расширением *.bds — содержит графическое изображение сгенерированного модуля;
- 3) файл, позволяющий поместить графическое изображение сгенерированного модуля в папку проекта.

К достоинствам подобных средств автоматизации можно отнести:

- сокращение времени и сложности процесса проектирования за счет возможности генерации и использования стандартных блоков программ, описывающие те или иные типы устройств;

- снижение требований к квалификации эксперта.

Однако данные средства имеют следующие недостатки:

- закрытость, неспособность изменить алгоритм работы ЭС в других условиях, т.е. отсутствует возможность обучения тем или иным методикам проектирования; таким образом, закрывается путь обычному пользователю САПР, обладающему низкой квалификацией в инженерии знаний, передать свою методику проектирования определенного вида устройств;

- эксперты-программисты, формирующие подобного рода программы автоматизации синтеза HDL-программ, обладают высоким уровнем квалификации в инженерии знаний, т.е. способны определить набор правил, входящих в методики проектирования, а также организовать диалог синтеза требуемого решения; таким образом, создание подобным путем программ автоматизации очень трудоемко и требует высокого уровня квалификации от разработчика;

- отсутствие возможности адаптации на другие уровни квалификации эксперта, а также на различные источники воплощения методик проектирования;

– отсутствует возможность построения вложенных шаблонов, таким образом, формируются архитектуры только базовых типов блоков, допускающие только одноуровневую декомпозицию.

Таким образом, актуальной является задача создания средств подобного типа, лишенных данных недостатков.

2. Уровни квалификации глупого эксперта

Общая квалификация экспертов в предметной области, выступающих в роли инженеров по знаниям, была описана в работе [4]. В данном исследовании рассматривается режим «глупого» эксперта, который не может самостоятельно формировать БЗ и перекладывает эту проблему на инструментальную систему, т.е. пользователь не в состоянии сформировать набор правил, составляющих методику проектирования, и не в состоянии организовать диалог ввода параметров желаемого устройства. С большой долей уверенности можно отметить, что большинство пользователей САПР относится именно к этому классу экспертов в инженерии знаний.

Классифицируем экспертов, «глупых» с точки зрения инженерии знаний, но уже с точки зрения знания языка VHDL, т.е. оценим квалификацию и уровень подготовки эксперта как проектировщика в данной предметной области. С этой точки зрения эксперт может иметь такие уровни квалификации:

- хорошо знать язык VHDL (умный);
- плохо знать язык VHDL (средний);
- вообще не знать язык VHDL (глупый).

Данная классификация рассматривает уровни квалификации эксперта с точки зрения владения методами и инструментальными средствами языка VHDL, а так же – с точки зрения наличия и уровня реализации его собственных разработок, т.е. VHDL-программ.

3. Классификация источников библиотек программ

Рассмотрим существующие источники библиотек-программ. Это может быть:

- пользовательские программы, написанные вручную, которые сам пользователь может характеризовать и складывать в структуру, отражающую его личную библиотеку;

- полученные путем автоматической перекодировки в язык VHDL [2] из наборов решений, подготовленных пользователем в САПР как модели структур (САПР типа OrCAD);

- библиотечные программы, предоставляемые пользователю в тех же САПР (сам САПР VHDL) для создания своих решений;

- сгенерированные пользователем существующими в САПР генераторами или мастерами в процессе создания своих решений [3].

Т.е. библиотеки программ можно классифицировать следующим образом: написанные вручную, перекодированные, подготовленные, сгенерированные.

Кроме того, множество библиотечных программ, которыми располагает пользователь и на основе которых строится база знаний ЭС, может иметь различную мощность по числу программ:

- программа может быть одна;
- их может быть много.

Библиотечные программы, независимо от их числа, могут иметь различную иерархию структуры модулей, отличаться числом типов используемых модулей, глубиной вложенности блоков.

Данные условия в своей совокупности могут определять, достаточна ли мощность предлагаемой библиотеки программ для создания полноценной БЗ или - не достаточна.

Данные условия могут рассматриваться как специфические требования в условиях, в которых создается база знаний, и по существу является аспектом проблемной адаптации инструментальной оболочки для создания интеллектуальной САПР.

4. Оценки эффективности функционирования базы знаний

Определим меры оценки эффективности возможного уровня автоматизации функционирования уже готового САПР для различных уровней квалификации, а также количества и сложности программ, предоставляемых для обучения. Предлагаются следующие оценки:

- уровень полноты - *число автоматически функционирующих модулей знаний*, формирующих устройства того либо иного типа по отношению к общему числу типов, используемых в заданной проектной задаче; возможные оценки: низкий процент, средний, высокий процент;

- *уровень автоматизации функционирования единичного модуля знаний*, который характеризуется числом и типами параметров, привлекаемых для синтеза данного типа изделий в рамках модуля знаний; возможные оценки: слабый (параметров мало и не все они имеют смысл), средний (большинство параметров имеют реальный смысл и число параметров среднее), хороший (число параметров максимально возможное, все они имеют смысл).

Рассмотрим, для примера, некоторые возможные оценки качества БЗ, исходя из условий создания БЗ и возможных методов автоматизации построения базы знаний как системы модулей знаний, связанных с различными типами блоков.

Библиотека VHDL-программ может иметь вид:

1) Одна программа, написанная вручную;

- можно выявить структуру программы как иерархию вложенных структурных блоков, т.е. онтологию программы (в подрежиме «умного» глупого эксперта);

2) Много программ, написанных вручную; это путь создания полноценной БЗ; т.е. – на выходе можно получить:

– систему полноценных модулей знаний (в подрежиме «умного» глупого эксперта); достигается полнота целевого пространства систем и пространства обликос систем (ЦПС/ПОС), полнота модели, проектной процедуры; обеспечивается наличие системы вложенных модулей;

– систему упрощенных модулей знаний (в подрежиме «глупого» глупого эксперта);

– и т.д.

Полный набор возможных условий создания базы знаний и ее оценки качества в рамках предложенной классификации приведен в таблице 1.

Таблица 1 – Влияние условий создания на качество БЗ

| № | Условия | | | | Оценки | | | | | | |
|---|--|--|------------------------|---|-------------------|--------------------|------------------------------|--------|-----|---------|--------|
| | Квалификаци я эксперта | Источни к знаний (библиот ек) | Кол-во програм м | Иерархичес кая сложность | Достато чность | Уровень полноты | Уровень автоматизац ии | | | | |
| 1 | хорошо знает язык VHDL (умный) | 1,2,3,4 | Одна | высокая | + | средний | хороший | | | | |
| | | | | средняя | +/- | средний | | | | | |
| | | | | низкая | - | низкий | | | | | |
| | | | Много | высокая | + | высокий | | | | | |
| | | | | средняя | + | средняя | | | | | |
| | | | | низкая | + | низкий | | | | | |
| 2 | плохо знает язык VHDL (средний) | 2,3,4 | Одна | средняя | +/- | низкий | средний | | | | |
| | | | | низкая | - | низкий | | | | | |
| | | | Много | средняя | + | средний | | | | | |
| | | | | низкая | + | средний | | | | | |
| | | | 3 | Вообще не знает язык VHDL (глупый) | 4 | Одна | | низкая | - | низкая | низкий |
| | | | | | | Много | | низкая | +/- | средняя | |

В таблице 1 источники знаний обозначены номерами 1,2,3,4, которые отвечают описанным выше классам библиотек: написанные вручную, перекодированные, подготовленные, сгенерированные.

Предлагаемые градации оценок получаемой БЗ выбраны, исходя из приведенной в данной статье градации условий создания БЗ.

Заключение

В работе выполнен анализ существующего уровня автоматизации генерации VHDL-программ в существующих САПР, проведена классификация существующих возможных источников знаний, определены уровни квалификации «глупого» эксперта и его возможности в наполнении БЗ, определены оценки эффективности функционирования базы знаний, соответствующей требуемой методике проектирования некоторого типа изделия.

Таким образом, возникает необходимость создания инструментальной оболочки для наполнения БЗ ЭС путем извлечения методик проектирования из набора прецедентов в разных условиях квалификации эксперта и имеющихся источников знаний (библиотек программ). Построение такого рода оболочки позволило бы существенно повысить эффективность процесса проектирования VHDL-программ обычными пользователями, т.е. – проектировщиками.

Список литературы

1. Григорьев А.В., Грищенко Д.А. Перспективные направления решения задачи синтеза HDL-программ в САПР РЭА.- Сборник трудов XI международной научной конференции им. Т.А. Таран, Киев, 2011. С. 75-81
2. Кузнецова С. Новые возможности OrCAD 10.0 [Текст] / С. Кузнецова. – 2007. – Режим доступа: <http://catalog.gaw.ru/index.php?page=document&id=1409>
3. Шалагинов А. Изучаем Active-HDL 7.1. Урок 6. Инструменты, повышающие эффективность создания HDL-моделей [Текст] / А. Шалагинов // Компоненты и технологии. – №8. – 2009. – С. 118-123.
4. Григорьев А.В., Морозова О.В. Построение двухсторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР. - Сборник трудов XI международной научной конференции им. Т.А. Таран, Киев, 2011. С. 68-75.

Получено 10.09.2011