

## ИНТЕЛЛЕКТУАЛИЗАЦИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ АППАРАТУРЫ СРЕДСТВАМИ ЯЗЫКА VHDL

Григорьев А.В., Кошелева Д.А.

ДонНТУ

grigorie@r5.dgtu.donetsk.ua

### **Abstract**

*Grigoriev A.V., Kosheleva D.A. Apparatus desing process intelektuation by language funds VHDL. Considers a intelektuation task CAD of apparatus designing on the basis of language VHDL by funds of specialized instrumental envelope in knowledges conception conditions on base of originative grammars with productions definite over them.*

### **Введение**

Язык проектирования аппаратуры HDL является основным инструментальным средством многочисленных САПР - Active HDL, OrCAD и т.д. [1-4]. HDL наряду с Verilog является одним из двух наиболее популярных средств проектирования аппаратуры в мире и - самым популярным в Европе.

Процесс проектирования аппаратуры с использованием языка HDL состоит в разработке программы, реализующей необходимый алгоритм функционирования проектируемого устройства. Имеющиеся компиляторы обеспечивают перевод алгоритмов на HDL в схемы, соответствующие выбранной аппаратной базе реализации, например - в ПЛИС. Представление проектов в форме текста в значительной мере свободно от недостатков графических и табличных форм представления проектов, таких как трудоёмкость процедуры ввода, трудность поиска ошибок. Вместе с тем, особо значимыми в этом случае становятся процедуры отладки и верификации проектных решений. К достоинствам описания проекта в текстовом виде (на языках Verilog, VHDL и др.) можно отнести [2]:

- компактность представления очень сложных логических проектов;
- легкость понимания и большая скорость создания проектов, включая относительную простоту поиска ошибок и внесения модификации в проект;
- лёгкость переноса проекта в различные приборные среды;
- простота настройки и перенастройки параметров разрабатываемых устройств или их фрагментов, например, разрядности или подмножества реализуемых функций;

- сближение методов проектирования аппаратуры и программных средств;
- улучшение взаимопонимания между проектировщиками различных подсистем вычислительных комплексов;
- возможность перевода программ пользователя, построенных на базе стандартных универсальных языков (например, С), в языки описания аппаратуры и обратно;
- наличие стандартных комментариев в текстах программ.

Главное достоинство языка HDL - это возможность накопления и повторного использования экспертных знаний об апробированных решениях в форме, удобной для хранения, передачи и редактирования. Однако, в настоящее время нет средств обучения САПР, позволяющих обеспечить процесс передачи экспертных знаний о методиках проектирования либо напрямую - от эксперта, либо на основе ряда примеров, т.е. накопленных апробированных решений. Обеспечение такой возможности позволит повысить эффективность использования САПР на базе языка HDL за счет повышения уровня автоматизации процесса проектирования.

Цель данной работы - описание средств обучения САПР, позволяющих обеспечить процесс передачи экспертных знаний о методиках проектирования путем создания на основе обучения интеллектуальной надстройки в среде специализированной оболочки для построения интеллектуальных САПР.

## **1. Анализ существующих методов автоматизации проектирования в САПР на базе HDL**

В настоящее время в САПР-ах 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. Среди имеющихся средств автоматизации создания моделей на языке VHDL в различных САПР можно назвать [2]:

- 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.

*Макросы* служат средством автоматизации интерфейса самой системы проектирования и интеграции её в проект создаваемых HDL программ. Они включают:

А) описания, заключаемые в системные ключевые слова, исходного набора файлов проекта;

Б) входные конкретные данные для программы;

В) время симуляции в системе.

Во время запуска макроса происходит передача параметров в систему проектирования. У разработчика появляются предпосылки расширить возможности языка за счёт введения собственных макроопределений, ориентированных на класс реализуемых проектов.

Набор подключаемых библиотек констант и функций, применяемых в текстах программ на VHDL, может быть как стандартным, так и определённым пользователем. В нём есть стандартизированные константы и готовые функции, применяемые в текстах программ на VHDL.

*Мастера* помогают быстро подготовить шаблон для заданной архитектуры программируемой системы и даже определить количество начальных параметров и типов. Но все остальные дальнейшие заботы по написанию кода программ ложатся, естественно, на программиста-разработчика.

Собственно текстовые модельные описания используется в современных САПР очень широко. Можно назвать, для примера, некоторые новые возможности, связанные с использованием текстов в САПР Active HDL 5.1 [5].

Данные возможности можно отнести к достоинствам САПР:

1) В просмотрщике структуры проекта:

- модификация (создание) текстового файла конфигурации VHDL;

2) В графических процессах:

- текстовое описание процессов;

3) В редакторе блок-схем:

- механизм синтеза кода на VHDL по построенной блок-схеме для любого элемента блок-диаграммы;

- предусмотрены комментарии ко всем элементам блок-схемы (отдельная вкладка), вводятся пользователем; комментарии автоматически входят в генерируемый код VHDL;

- генерация по блок-диаграмме архитектуры и сохранение ее в проекте и т.д.

Вместе с тем, к типичным недостаткам САПР Active HDL 5.1 можно отнести:

- отсутствие явного языка формирования технического задания;

- имеющиеся комментарии могут рассматриваться лишь как фрагменты ТЗ.

## **2. Постановка задачи**

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 маленькой специализированной САПР, ориентированной на некоторый класс решений (устройств). Разработка каждого нового мастера предполагает творческий процесс программирования, включая:

- разработку диалоговых шагов, включая определение параметров проектирования и их возможных значений;
- определение фрагментов текстов, соответствующих тому или иному набору значений параметров требуемого решения, определяемому в диалоге с пользователем;
- разработку алгоритма компоновки определенных фрагментов текстов по заданному набору значений параметров.

Перед разработчиком встает задача разработки языка ТЗ, определение состава возможных решений и построение алгоритма выбора нужного решения по заданному набору значений параметров. Т.о., разработка такого мастера есть форма передачи разработчиком своего опыта по проектированию некоторого класса устройств.

Недостатки такого пути:

- сложность процесса передачи знаний;
- необходимость отладки и верификации мастеров.

Таким образом, данный 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.

### **3. Предлагаемый путь создания интеллектуальной надстройки**

#### **3.1. Общая модель интеллектуальной надстройки**

Для решения задачи предлагается использовать путь создания интеллектуальной надстройки над VHDL, описанный в [6-9]. Основанием такого выбора является возможность учесть как достоинства, так и недостатки САПР на базе VHDL. По классификации, приведенной в [7], любая САПР на базе VHDL, как конкретный проблемно-ориентированный САПР имеет следующие характерные черты:

- САПР может обеспечить для обучения базы знаний совокупность апробированных решений, представленных в виде текстов,
- САПР не имеет явно выраженного языка задания ТЗ.

Соответственно, возможны следующие пути построения интеллектуальной надстройки, являющиеся вариантом модели САПР решения типичных задач проектирования:

I) Создание целевого пространства систем (ЦПС), т.е. пространства возможных решений-прототипов.

Имеется два пути:

- A) САПР может быть источником решений-прототипов; далее ЦПС в нужной форме представления создается инструментальной оболочкой автоматически;
- B) Эксперт в данной предметной области может самостоятельно ввести ЦПС в нужной форме представления;

II) Создание пространства обликос систем (ПОС), т.е. пространства технических заданий.

Имеется три пути:

- C) ПОС как совокупность слов-ТЗ может быть сформировано оболочкой автоматически на основе имеющегося ЦПС, заданного в требуемой форме представления.
- D) Источником множества слов-ТЗ для создания ПОС может быть только пользователь - эксперт в предметной области, в силу отсутствия в САПР языка описания ТЗ. В этом случае ПОС в нужной форме представления формируется автоматически.
- E) Эксперт в данной предметной области может самостоятельно ввести ПОС в требуемой форме представления.

Главная цель функционирования  $K_0$ , составляющая главный признак множества решений в ПОС, соответствует классу проектируемых устройств и задается в этом случае неявно. Построенная таким образом экспертная система может выступать в качестве надстройки над САПР VHDL. При этом САПР может выступать в двух качествах:

- 1) Как источник семантически верных решений, предназначенных для обучения базы знаний;
- 2) Как приемник синтезированных в базе знаний решений для последующей их интерпретации и документирования.

### **3.2. Модули знаний как отражение структурно-функциональной модели аппаратуры в VHDL**

В соответствии с [8] для организации базы знаний используется модульный принцип. Модуль знаний в инструментальной оболочке для построения интеллектуальных САПР формируются в форме порождающей контекстно-свободной грамматики, эквивалентной И-ИЛИ-дереву [5], с определенным над ней множеством продукций, явных или неявных. Грамматика описывает множество синтаксически верных (возможных) решений, а продукции, как явные так и неявные, определяют множество правил вывода, описывающих вычленение семантически верного подмножества решений, соответствующего некоторому набору требований, т.е. продукции описывают логику проектирования, исходя из требований к решению. Специфика формирования модулей знаний в

данном случае определяется спецификой языка VHDL. Модель устройства на языке VHDL имеет две явно определенные стороны описания: структурную и функциональную.

Структурная часть - это:

1. Совокупность блоков - процессов, включая их внешние полюса (входы-выходы),
2. Список связей между блоками.

Функциональная часть - это:

1. Тексты описания процессов, составляющих базовые блоки структуры;
2. Задание значений переменных – полюсов, а так же описание моделей сигналов.

Рассмотрим возможные способы формирования модулей знаний, отличающиеся уровнем знания о грамматике языка VHDL в среде инструментальной оболочки. Минимальный уровень знаний о языке – это способность оболочки выделить значимую часть текста и – незначимую. К незначимой части можно отнести комментарии и лишние пробелы. Глубина знаний о процедурном делении языка VHDL предполагает, что отдельный модуль знаний может задавать множество решений, относящихся к таким классам: 1) проекты одного типа в целом; 2) архитектуры одного типа; 3) процессы одного типа.

В каждом случае пара ЦПС – ПОС строится для различного числа модулей знаний. Наиболее сложный случай предполагает построение модуля знаний для отдельного типа блоков из блок-схемы, рассматриваемого на любом структурном уровне. В текстовой модели на языке VHDL в этом случае должно быть явно выделено:

- идентификатор модели, т.е. блока;
- модель внешней границы как совокупности входов-выходов, т.е. свойств внешней границы блока;
- модель внутренней границы блока, как идентифицируемая совокупность внутренних переменных (инкапсулированные свойства метода);
- внутренняя модель блока, функциональная или структурная.

Все четыре компонента нуждаются в собственной нумерации прототипов. Это позволит рассматривать подмножества:

- моделей блоков  $B_i$ ;
- моделей внешней границы  $G_{ij}$ , соответствующих типу блока  $B_i$ ;
- моделей внутренней границы блока  $V_{kj}$ , соответствующих внешней границе  $G_{ij}$ ;
- моделей функции или структуры блока, соответствующих внутренней границе  $V_{kj}$ .

Данный подход был ранее описан в [9]. Наличие таких текстовых фрагментов позволит создать модуль знаний, стыкуемый с другими модулями знаний на уровне списка внешних свойств блока.

### **Заключение**

В работе изложены следующие результаты: 1) анализ средства автоматизации проектирования аппаратуры в среде языка VHDL; 2) определение возможных путей создания интеллектуальной надстройки над САПР VHDL в среде специализированной инструментальной оболочки; 3) анализ специфики языка VHDL с точки зрения формирования модулей знаний.

### **Литература**

1. The VHDL Cookbook First Edition. July, 1990 Peter J. Ashenden Dept. Computer Science University of Adelaide South Australia © 1990, Peter J. Ashenden.
2. Бибило П.Н. Основы языка VHDL. - Минск: Ин-т техн. кибернетики НАН Беларуси, 1999. - 202 с., ил.
3. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики – СПб.: БХВ-Петербург, 2002. – 608с.: ил.
4. Е.З. Перельройзен. Проектируем на VHDL – М.: СОЛОН-Пресс, 2004 – 448 с.: ил.-(Серия «Библиотека профессионала»).
5. Продукты фирмы Aldec, Inc., [www.aldec.com.ua](http://www.aldec.com.ua).
6. Григорьев А.В. Семиотическая модель базы знаний САПР. Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: - Донецк, ДонГТУ, 1999. - С. 30-37.
7. Григорьев А.В. Уровни квалификации экспертов и пути создания интеллектуальных САПР. В этом же сборнике.
8. Григорьев А.В. Принципы организации вывода решений в базе знаний инструментальной оболочки для создания интеллектуальных САПР. // Практика і перспективи розвитку інституційного партнерства». Вісник ДонГТУ – ТРТУ. Донецьк: РВА ДонНТУ, 2003 – С. 96-106.
9. А.В. Григорьев. Содержание некоторых категорий абстракций в теории интеллектуальных САПР. Материалы XVIII-й международной конференции "Интеллектуальные САПР - 2003" CAD-2003. Дивноморское, 3-10 сентября 2003, в 2-х томах. - Таганрог: Из-во ТРТУ. - Т.2., С. 11-18.

Дата надходження до редакції 06.04.2005 р.