

\* \* \* \* \*

**Введение.** Проектирование систем на поведенческом уровне позволяет снижать как размерность решаемых задач, так и требования к моделированию проектов на более детальных и трудоемких уровнях проектирования. Это особо актуально при проектировании цифровой техники, так как степень интеграции микросхем и соответственно сложность и стоимость проектов делает проблемным детальное моделирование проектов на нижних уровнях детализации за приемлемое время и цену, что препятствует должной степени автоматизации.

Изобразительные средства для проектирования на поведенческом уровне предусматривают инструментальный описания процессов, развивающихся параллельно-последовательно, что в свою очередь требует поддержки соответствующих аппаратных либо программных средств.

Здесь рассматриваются проблемы организации системы моделирования, где объектом моделирования является проект, описанный заказчиком на поведенческом уровне с помощью программных компонентов типа entity, architecture языка VHDL (стандарты в области автоматизации проектирования VHDL'93 - ANSI/IEEE Std 1076-1993, VHDL-AMS - Std 1076.1-1999) [1–5]. Компоненты типа entity описывают входы и выходы устройств как порты проектов, а компоненты типа architecture – алгоритмы преобразования входных объектов (сигналов) в выходные объекты (сигналы). Последние компоненты с помощью механизма процессов VHDL представляются набором функций, реализуемых проектом, и записываются в терминах последовательных и параллельных операторов. Взаимодействие функций отображается через сигналы, к которым они чувствительны и которые они обрабатывают.

**Характеристики системы.** Система должна обеспечивать имитацию функционирования проекта, описанного на языке VHDL, управляемую трассой входных сигналов, соответствующей временной диаграмме работы проектируемого устройства.

Система должна функционировать в разных режимах, включая: - режим отладки (этап формирования описания закона функционирования проекта с целью его синтаксической, семантической проверки); - режим верификации описания проекта на эталонных наборах данных; - режим моделирования, оценки характеристик проекта и степени их соответствия заданным критериям, ограничениям; - режим генерации тестовых наборов для последующего моделирования проекта на более детальных уровнях описания.

Система должна обеспечивать необходимую информационную поддержку обработки проектов (с учетом иерархичности проектов; степени детализации и версий описания проектов; интерфейсов проектов, задаваемых составом, типами, режимами использования сигналов портов и т.д.), соответствующую поддержку библиотек формирователей сигналов, эталонных наборов данных (с учетом статического, динамического режимов моделирования проектов).

**Структура и функционирование системы.** Эффективность системы может быть обеспечена рядом факторов. В том числе:

- промежуточным приведением исходных описаний проектов к некоторому стандартному однородному описанию (здесь процессному);
- выбором в качестве средства внутреннего представления описаний проектов языка высокого уровня (ЯВУ) с развитыми вычислительными средствами, возможностью управления парал-

лельными процессами и эффективным транслятором;

- преобразованием однородных описаний проектов в адекватные в функциональном отношении тексты на языке высокого уровня с последующим получением загрузочных модулей.

Соответственно этому система строится как совокупность лингвистического (входной и внутренние языки), информационного и программного обеспечения. Информационное обеспечение включает библиотеки исходных, промежуточных и ЯВУ-описаний проектов, библиотеки формирователей сигналов, библиотечные средства обработки компонентов моделей, включая обработку битовых векторов сигналов, библиотеку шаблонов функций, настраиваемых на проект и используемых в процессе моделирования. Программное обеспечение включает подсистему обработки описаний проектов, подсистему управления формирователями сигналов, подсистему генерации внутренних, исполнимых моделей проектов и подсистему управления имитационным моделированием проектов. Укрупненная структура системы приведена на рис. 1.

На вход системы поступают: - описания законов функционирования проектов на языке VHDL; - эталонные наборы данных для функционирования проекта; параметры, определяющие режимы моделирования.

Общая схема функционирования системы иллюстрируется на рис. 2 и предусматривает следующие этапы. На первом этапе, проводимом однократно, для каждого проекта по результатам анализа исходного описания создается ЯВУ-текст соответствующего программного компонента, функционально-адекватного исходному описанию. Полученное описание используется для проведения моделирования, сохраняется в библиотеке для последующего использования. На втором этапе, выполняемом многократно, производится имитационное, событийное моделирование описания проекта, загрузочный код которого получают транслятором ЯВУ.

Само моделирование сводится к порождению порядка иницирования блоков и процессов в блоках проекта в соответствии с призами их активности и алгоритмом функционирования проекта и к фиксации значений сигналов и времени, когда происходит их изменение. Этапы выполняются друг за другом либо автономно, что обеспечивает иерархический подход к моделированию и нисходящий, восходящий подходы к проектированию.

Подсистема обработки описаний проектов [6] обеспечивает анализ исходных описаний и заполнение библиотеки промежуточных описаний проектов в виде соответствующих информационных таблиц, структур данных ЯВУ.

Подсистема генерации и настройки моделей обеспечивает автоматизацию построения имитационных моделей проектов на языке высокого уровня по их исходным, процессным описаниям [6, 8]. Исходными данными для подсистемы служит библиотека промежуточных описаний проектов. Представляет собой набор взаимосвязанных структур данных ЯВУ, описывающих компоненты и элементы компонентов VHDL-проектов произвольного стиля. Это структуры, отображающие декларативную информацию (описания интерфейсов, сигналов и т.п.), и структуры, отображающие архитектуру проектов в виде составляющих их параллельных и последовательных операторов. Функциональность операторов раскрывается через описание составляющих их выражений, представляемых деревьями термов (информация хранится в виде соответствующих списковых структур).

*Муравьев Геннадий Леонидович, к.т.н., профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета.*

*Мухов Сергей Владимирович, доцент кафедры интеллектуальных информационных технологий Брестского государственного технического университета*

*Шуть В.Н., к.т.н., доцент кафедры интеллектуальных информационных технологий Брестского государственного технического университета.*

*Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.*

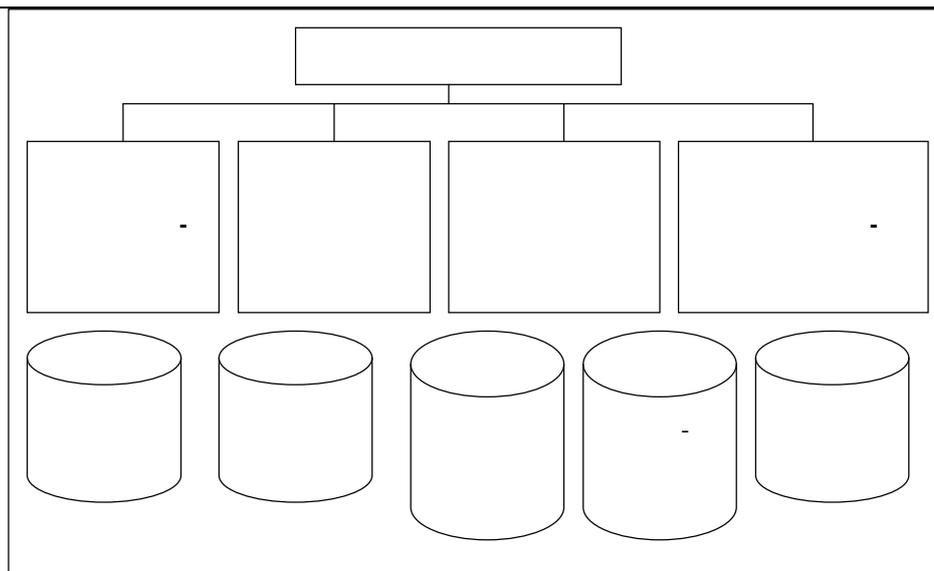


Рис. 1. Структура системы

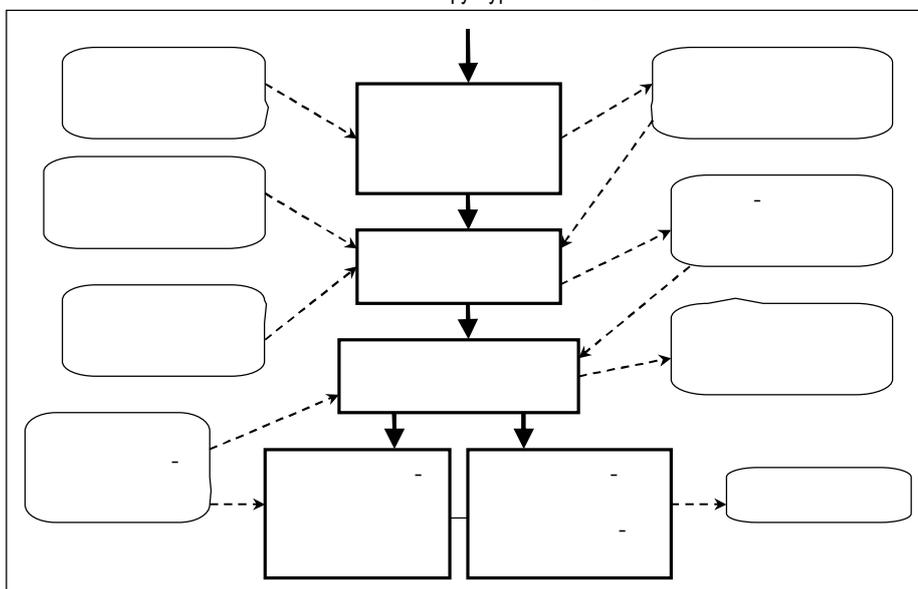


Рис. 2. Порядок функционирования системы

Подсистема обеспечивает получение имитационных моделей в виде ЯВУ-описаний, функционально адекватных проектам, и готовых для последующего исполнения. При этом в процессе генерации моделей к соответствующему ЯВУ-описанию проекта подключаются необходимые для работы модели средства, функции библиотеки обработки битовых векторов, а также шаблоны программ управления, настраиваемые на интерфейс конкретного проекта. Программы используются для управления моделированием на уровне описанных в блоке процессов посредством их запуска и синхронизации.

Построение моделей выполняется путем анализа описаний проектов и обработки их по заранее установленным правилам, что включает перевод произвольных описаний в процессные с последующей трансформацией процессных описаний и их операторов в результативные, модельные описания в терминах ЯВУ.

Подсистема включает модуль построения процессного описания проекта, набор модулей для анализа описаний интерфейсов, архитектур проектов, подпрограмм и функций проекта, а также модули генерации текстов в терминах ЯВУ. Последние модули базируются на многократном применении генераторов кодов для параллельных и последовательных операторов проекта, использующих рекурсивно

анализаторы выражений, обрабатывающих соответствующие списковые структуры.

Специализированная библиотека средств обработки битовых векторов [9] поддерживает программно векторное представление сигналов и операций над ними, что является одним из средств повышения компактности описаний моделей проектов, скорости моделирования. Указанная библиотека предназначена для обеспечения эффективной поддержки операций над объектами типа `bit_vector`, составляющими вместе с типами `bit`, `boolean` основу predefined двоичной арифметики VHDL [1–5].

Для моделирования битовых векторов произвольной длины используются специализированные структуры данных - упакованный формат. Они включают массивы элементов базового типа ЯВУ для хранения значений и служебные структуры, описывающие область информации, формат хранения данных и другую служебную информацию, что обеспечивает мобильность, самоопределение данных и реализацию пересылок данных путем простого изменения описательных структур. Кроме этого, библиотека обеспечивает: - корректное взаимодействие разнотипных объектов (например, `bit`, `boolean`, `integer`, `array of bit`, `array of boolean`); - эффективное преобразование типов объектов; - минимальное по времени выполнение операций за

счет использования соответствующих машинных команд и минимизации числа операций преобразования типов данных; - реализацию формул с использованием "смесей" битовых векторов произвольной длины, битовых и литеральных значений; - возможность контроля результатов выполнения операций, изменения диапазона значения, ситуации потери значимости, переполнения результатов; - поддержку аппарата предопределенных атрибутов VHDL.

Подсистема формирователей сигналов обеспечивает все виды работ с формирователями, драйверами сигналов VHDL - специализированными структурами данных, отображающими характер изменения сигналов во времени и являющимися источниками их значений при моделировании [7]. Поддерживает весь комплекс работ, связанных с подготовкой входных воздействий, их использованием для верификации проекта моделированием, оценки реакций моделируемого проекта. Исходными данными для подсистемы служат: входные воздействия, имитирующие реальное окружение моделируемого проекта, выходные и эталонные реакции проекта, интерфейсы проектов и пакеты заказчика, описывающие пользовательские типы сигналов. Подсистема обеспечивает настройку программных модулей системы на интерфейсы моделируемых проектов, выполняет подготовку входных воздействий проектов, а в ходе проведения моделирования обеспечивает подачу входных сигналов в имитационную модель проекта, а также накопление и анализ реакций проекта на входные сигналы.

Организация формирователей производится с учетом их привязки к проектам, портам, версиям архитектурной реализации проекта, с учетом требований к функциональности атрибутивной арифметики и набору типов, допустимых синтаксисом и семантикой VHDL, а также с учетом поддержки статического и динамического способов представления входных воздействий. Процесс автоматизированной обработки формирователей при моделировании строится как результат взаимодействия библиотеки формирователей и соответствующих функциональных модулей.

Библиотека формирователей сигналов включает: - логическое оглавление (перечень проектов, перечень соответствующих им структур наборов данных и структур эталонных наборов данных); - библиотеку наборов данных; - библиотеку эталонных наборов данных; - библиотеку выходных реакций.

Подсистема поддерживает взаимодействие пользователей с модулями, внутреннюю обработку формирователей, обеспечивает их эффективное взаимодействие с подсистемой имитационного моделирования и включает ряд модулей. Это модуль, обеспечивающий пользовательский интерфейс для выбора, просмотра, редактирования, создания, удаления и выполнения других операций с наборами данных, формирователями, их отдельными полями. Модуль, конвертирующий наборы данных в формат, обеспечивающий эффективное моделирование за счет выполнения предварительных операций по идентификации значений сигналов. Модуль, обрабатывающий результаты моделирования, обеспечивающий расчет характеристик, просмотр трасс моделирования. Модуль, реализующий интерфейс взаимодействия формирователей и модели, обеспечивающий эффективное проведение моделирования проектов. Модуль предназначен для определения текущих значений входных сигналов и подачи сигналов в систему моделирования через порты проекта, он выполняет чтение значений сигналов из портов проекта и фиксацию текущих значений выходных сигналов, определяет предопределенные атрибуты сигналов. Часть модулей функционирует автономно, часть используется в соответствии с алгоритмом организации подсистемы имитационного моделирования.

Подсистема моделирования осуществляет управляемое событиями имитационное моделирование проекта. Предварительно подсистема настраивается на интерфейс моделируемого проекта для обеспечения работы с формирователями сигналов (подсистема управления формирователями сигналов). После настройки создается загрузочный модуль, представляющий сборку необходимых шаблонных модулей подсистемы моделирования и ЯВУ-текста проекта. При его запуске иницируется управляющая программа, определяющая порядок иницирования процессов в соответствии с алгоритмом функционирования проекта, и осуществляется обработка сигналов рядом модулей, реализующих элементарные функциональные преобразования.

**Заключение.** В работе рассмотрены проблемы организации системы моделирования проектов устройств, ориентированной на реализацию имитационных расчетов по процессным описаниям проектов. Система предназначена для обеспечения на единой технологической и информационной основе задач оценки корректности описания законов функционирования проектов, прогнозирования их характеристик в ходе реализации проектных процедур, оценки степени их соответствия критериям эффективности.

Применительно к языку VHDL рассмотрена структура информационного, программного, лингвистического обеспечения системы, состав подсистем и их взаимодействие в ходе моделирования. Сформулированы требования к характеристикам подсистем и их обеспечению. Полученные результаты могут применяться для построения подсистем, используемых в системах автоматизированного проектирования верхнего уровня, либо автономно.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. VHDL для моделирования, синтеза и формальной верификации аппаратуры; пер. с англ. - М.: Радио и связь, 1995. - 360 с.
2. Бибило, П.Н. Синтез логических схем с использованием языка VHDL. - М.: СОЛОН-Р, 2002. - 384 с.
3. Бибило, П.Н. Основы языка VHDL. - М.: СОЛОН-Р, 2000. - 210 с.
4. Сергиенко, А.М. VHDL для проектирования вычислительных устройств. - К.: Корнейчук, 2003. - 208 с.
5. Поляков, А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. - М.: СОЛОН-Пресс, 2003. - 320 с.
6. Муравьев, Г.Л. Интерпретация VHDL-описаний, согласованная с процессным способом моделирования / Г.Л. Муравьев, В.Н. Шуть // Вестник БрГТУ. - 2005. - № 5(35): Физика, математика, информатика. - С. 81-84.
7. Мухов, С.В. Организация библиотеки формирователей сигналов для моделирования проектов цифровых устройств / С.В. Мухов, Г.Л. Муравьев // Вестник БрГТУ. - 2008. - № 5 (53): Физика, математика, информатика. - С. 98-100.
8. Муравьев, Г.Л. Построение моделей по описаниям, согласованным с процессным способом моделирования / Г.Л. Муравьев, В.И. Хвещук // Современные информационные компьютерные технологии mcIT-2008: сб. научн. ст. в 2 ч. / Гродно: ГрГУ, 2008. - Ч. 2. - С. 235-238.
9. Мухов, С.В. Компьютерная поддержка специализированных структур данных / С.В. Мухов, Г.Л. Муравьев // Технологии информатизации и управления TIM-2009: сб. научн. ст. в 2 ч. - Гродно: ФГУО ИТИУ БГУ, 2009. - Ч. 2. - С. 59-62.

28.10.10

#### MURAVJOV G.L., MUHOV S.V. Automation of simulation modeling of behavioral descriptions of projects

The approach to the organization of system of modeling is considered. The system structure oriented on modeling of behavioral descriptions of projects is resulted.