

Московский Государственный Технический Университет

им. Н. Э. Баумана

На правах рукописи

Котельницкий Алексей Владимирович

УДК: 629.7.05

**ПРОГРАММНАЯ ПОВЕДЕНЧЕСКАЯ МОДЕЛЬ БЛОКА ИНЖЕКЦИИ ЯДРА
МАРШРУТИЗАТОРА**

А в т о р е ф е р а т

**Квалификационной работы бакалавра по направлению 210200
Проектирование и технология электронных средств**

Москва – 2012

Работа выполнена в Московском Государственном Техническом Университете

им. Н. Э. Баумана

Научный руководитель:

Л.В.Журавлева
канд. техн. наук, доцент кафедры ИУ4
«Проектирование и технология производства
ЭА» МГТУ им.Н.Э.Баумана

Научный консультант:

А.И.Власов
канд. техн. наук, доцент кафедры ИУ4
«Проектирование и технология производства
ЭА» МГТУ им.Н.Э.Баумана

Ведущее предприятие: ОАО "НИЦЭВТ" (г. Москва)

Защита квалификационной работы бакалавра состоится 29 июня 2012 года на заседании Государственной аттестационной комиссии по направлению 210200: «Проектирование и технология электронных средств» в Московском Государственном Техническом Университете им. Н.Э.Баумана (ауд.278).

Ваши отзывы в двух экземплярах просьба высылать по адресу:
105005, г.Москва, 2-ая Бауманская ул., д.5, ИУ-4.

Автореферат разослан «__» _____ 2012 г.

Ученый секретарь Государственной квалификационной комиссии по направлению 210200: «Проектирование и технология электронных средств»
Доцент, кандидат технических наук Лавров А.В.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Работа посвящена исследованию принципов построения программных поведенческих моделей логических блоков встраиваемых систем. В работе подробно рассматривается разработка поведенческой модели блока инъекции маршрутизатора высокоскоростной отказоустойчивой коммуникационной сети (ВОКС) «Ангара».

Объектом исследования является блок инъекции ядра маршрутизатора ВОКС «Ангара». Блок инъекции является одной из ключевых составляющих маршрутизатора, обеспечивающих формирование и маршрутизацию целостного пакета данных для отправки в сеть.

Актуальность работы определяется возможностью повторного применения программных поведенческих моделей при использовании спиралевидной модели проектирования систем на кристалле. Данная методология предполагает проектирование одновременно на нескольких уровнях абстракции: на уровне функционального представления системы, регистровых передач и логическом уровне. При ее использовании функциональное представление проекта в виде поведенческой модели, может быть использовано на всех уровнях.

При разработке программных поведенческих моделей внутренняя структура схемы не учитывается. Разработанные модели затем используются для решения различных задач – оценки архитектуры проектируемого устройства, верификации, а также для разработки и отладки тестов, другого программного обеспечения на ранних стадиях проектирования готовой системы на кристалле.

Функциональный уровень абстракции представляется поведенческой моделью, уровень регистровых передач – регистровой или RTL(register transfer level) моделью. Регистровая модель создается на языках описания аппаратуры интегральных схем, обычно Verilog или VHDL. Из такой модели синтезируются логические элементы цепи. Весь проект разбивается на части, и реализуется поэтапно, проходя все этапы проектирования системы на кристалле на каждой стадии проекта.

В рамках разработки маршрутизатора «Ангара» можно выделить следующие области применения поведенческой модели:

- архитектурное моделирование маршрутизатора;
- верификация;
- отладка программного обеспечения.

Методология спиралевидного проектирования предполагает разбиение процесса разработки устройства на несколько частей. Сначала в проекте выделяют базовую часть – набор функций, необходимых для выполнения основных задач устройства. Далее,

устройство с базовым функционалом проходит все этапы проектирования системы на кристалле. После завершения разработки базовой части, устройство усложняется и снова проходит все этапы проектирования. При этом, перед усложнением функционального наполнения, технические требования, изначально предъявляемые к устройству, могут быть пересмотрены.

Поскольку модель спиралевидного проектирования предполагает изменения технических требований в процессе проектирования системы на кристалле, одним из применений программной поведенческой модели является архитектурное моделирование. К примеру, при пересмотре технического задания, может возникнуть необходимость в реализации дополнительного логического блока. Для того чтобы избежать возможных ошибок, перед созданием кода регистровой модели проводится предварительное моделирование его работы. С этой целью разрабатывается поведенческая модель, которая отражает архитектуру и функциональность блока. Такая модель не содержит детального описания внутренней структуры блока и обычно разрабатывается на высокоуровневых языках программирования (C, C++). При внесении изменений в уже разработанные блоки предварительное моделирование позволит оценить и оптимизировать вносимые изменения.

Разработанная модель, в дальнейшем, может применяться в процессе верификации регистровой модели. В этом случае программная поведенческая модель используется в качестве эталонной (golden model). Для проверки соответствия RTL кода техническим требованиям, разрабатывается тестовое окружение, которое генерирует тестовые воздействия и подает их на вход поведенческой и верифицируемой модели. Затем результат работы двух моделей сравнивается и делается вывод о правильности работы RTL модели. Для синхронизации работы двух моделей используется библиотека описания аппаратуры SystemC, позволяющая запрограммировать потактовую работу сигналов и шин данных. Недостатком использования данной библиотеки является низкая скорость выполнения кода, по сравнению с C++.

Другим применением программных поведенческих моделей в процессе проектирования систем на кристалле (СнК) является отладка программного обеспечения (ПО). После архитектурного моделирования устройства выделяются подзадачи, выполняемые аппаратными и программными средствами. В результате проект разбивается на две части: программную и аппаратную. Обычно для отладки ПО целей используют ПЛИС или прототип кристалла. Однако это возможно только на последних стадиях проектирования, когда в распоряжении разработчиков уже есть отлаженная регистровая модель, из которой можно было бы сгенерировать код для программирования ПЛИС. Для

сокращения сроков выпуска изделия на первых этапах проектирования СнК в качестве виртуального кристалла используют поведенческую модель.

Постановка задачи:

Для сокращения сроков проектирования систем на кристалле функциональное представление логических блоков, представленных программными поведенческими моделями, может быть многократно использовано при различных этапах разработки: верификации, архитектурном моделировании и отладке программного обеспечения. Однако не всегда повторное использование разработанного программного кода возможно – разные задачи требуют разной степени детализации созданного программного описания. Решением данной проблемы может быть разработка универсальной поведенческой модели, состоящей из отключаемых компонентов: каждый компонент позволяет применять поведенческую модель для той, или иной задачи. Таким образом, выбор необходимого набора компонентов модели позволит применять разработанный код для различных задач в рамках маршрута проектирования системы на кристалле. Основными компонентами модели будут являться: функциональное ядро, блок учета временных задержек, интерфейсный блок для связи с RTL кодом. Поставленная задача может быть решена с использованием высокоуровневого языка программирования C++ и библиотеки описания аппаратуры SystemC, которая позволит смоделировать работу сигналов и осуществить связку между C++ и языком регистровых моделей Verilog.

Целью работы является создание универсальной программной поведенческой модели блока инъекции ядра маршрутизатора ВОКС «Ангара» на языке C++ с использованием библиотеки описания аппаратуры SystemC.

Для достижения поставленной задачи в работе **решены следующие задачи:**

- анализ существующего RTL представления блока инъекции маршрутизатора;
- разработка универсальной поведенческой программной модели блока инъекции маршрутизатора;
- разработка тестового окружения, формирующего входные тестовые воздействия для отладки разработанной модели на языке C++ с использованием библиотеки описания аппаратуры SystemC;
- исследование влияния структуры поведенческой модели на скорость ее моделирования.

Разработка программной поведенческой модели ведется с учетом основных применений данной модели в рамках проектирования маршрутизатора: архитектурное моделирование, отладка программного обеспечения и верификация.

Методы, используемые для решения поставленной задачи: методы маршрутизации ВОКС с глобально адресуемой памятью, методы реинжиниринга и управления проектами, методы объектно-ориентированного проектирования информационных систем.

Научная новизна работы:

- проведена систематизация и обобщение методов применения поведенческих моделей в рамках проектирования маршрута проектирования системы на кристалле, выявлены основные проблемы их применения, заключающиеся в вынужденной адаптации исходного кода под конкретные задачи и, таким образом, увеличением стоимости разработки;
- разработана оригинальная методика построения универсальных поведенческих моделей логических блоков систем на кристалле, позволяющая разрабатывать и эффективно применять подобные модели для архитектурного моделирования, верификации и отладки программного обеспечения;
- разработана методика функционального тестирования универсальных поведенческих моделей, заключающаяся во вложенной структуре тестового окружения, обеспечивающей проверку всех компонентов разработанного кода.

Практической ценностью работы является увеличение эффективности проектирования встраиваемых систем за счет уменьшения временных и экономических затрат на программирование дополнительных программных моделей. Применение универсальных программных поведенческих моделей при моделировании работы системы на кристалле позволяет увеличить оцениваемый временной отрезок работы ядра по сравнению с методикой применения RTL описания.

Апробация работы:

Материалы квалификационной работы были представлены на 14-ой международной конференции «Наукоемкие технологии и интеллектуальные системы - 2012».

Результатом работы является разработанная поведенческая программная модель блока инъекции маршрутизатора, функционально идентичная аналогичной регистровой модели. Конфигурация созданного программного кода позволяет использовать модель для верификации, архитектурного моделирования и отладки других программных компонентов системы не прибегая к его дополнительному изменению.

Реализация результатов:

Поведенческая модель блока инъекции позволяет проводить его верификацию и таким образом выявить ошибки, которые трудно обнаружить во время моделирования работы всей сети, что использовано в ОАО «НИЦЭВТ» при реализации проекта суперкомпьютерной сети «Ангара». Внедрение результатов работы продолжается.

Разработанная модель может быть использована при проектировании сложно-функциональных блоков систем на кристалле для архитектурного моделирования и отладки программного обеспечения проектируемых систем.

Результаты работы использованы в рамках курса «Архитектура и системотехника ЭВМ, комплексов и систем», читаемых на кафедре «Проектирование и технология производства электронной аппаратуры» МГТУ им.Н.Э.Баумана.

Достоверность полученных результатов подтверждена результатами тестирования разработанной модели в составе регистровой модели маршрутизатора ВОКС «Ангара». Проведенное тестирование показало стабильность работы и выполнение моделью всех обозначенных функций. Тактовая частота виртуального кристалла (количество смоделированных тактов за одну секунду реального времени) при верификации составила 3 кГц, что в 10 раз выше данного показателя у регистровой модели. Интеграция поведенческой модели в RTL описание показало увеличение скорости моделирования сети в 1.6 раз, что подтверждает эффективность применения разработанной модели.

Положения, выносимые на защиту:

- применение программных поведенческих позволяет повысить эффективность разработки систем на кристалле;
- разработанная программная поведенческая модель блока инъекции маршрутизатора «Ангара» позволяет увеличить скорость моделирования работы коммуникационной сети
- разработанное программное обеспечение позволяет проводить верификацию блока инъекции маршрутизатора «Ангара»

Публикации:

По материалам и основному содержанию работы опубликованы 2 печатные работы в трудах конференций и журнале ВАК.

Структура и объем работы:

Бакалаврская квалификационная работа состоит из введения, четырех глав и заключения. Общий объем работы 75 страниц, 23 рисунка, список использованных источников из 9 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность решения задач разработки поведенческих моделей при проектировании систем на кристалле, сформулированы цели и задачи исследования, обоснована научная новизна и изложена структура работы.

В первой главе рассматривается структура и основные принципы работы маршрутизатора «Ангара». Маршрутизатор предназначен для организации обмена данных между узлами коммуникационной сети. Подробно рассмотрен состав маршрутизатора и работа его компонентов. Основными компонентами маршрутизатора являются блоки приема и передачи (БПП), коммутатор, блок инъекции (БИ), блок эжекции (БЭ), блок управления интерфейсом (БУИ) и интерфейс с вычислительным узлом (ИВУ). Также рассмотрен алгоритм передачи данных по сети. Изложен формат передачи данных по сети: информация передается по пакетам, состоящих из флит – фиксированного набора данных, размер которых определяется шириной канала. Отдельное внимание уделено назначению служебных флит пакета. Подробно рассмотрена задача инъекции данных в сеть. Основной задачей инъекции является подготовка пакета данных для передачи в сеть и его маршрутизация. Алгоритм инъекции пакетов реализуется в блоке инъекции маршрутизатора. Структура и принципы работы блока инъекции указаны в третьем разделе первой главы. Помимо своего непосредственного предназначения блок инъекции также реализует передачу пакетов из сети в блок эжекции с предварительной проверкой целостности принятого пакета.

С целью выявления требований, предъявляемых к поведенческим моделям проводится анализ маршрута проектирования СнК и выявляются области применения поведенческих моделей. Отдельно рассматривается применение поведенческих моделей при верификации блоков устройства. С целью повышения эффективности разработки СнК ставится задача разработки универсальной модели, которую можно было бы применять на различных этапах проектирования. В ряде случаев, универсальная поведенческая модель, обладающая большой применяемостью имеет неоправданно низкую скорость моделирования. Таким образом, появляется техническое противоречие между применяемостью модели и скоростью моделирования. В тексте главы предлагается оригинальная структура поведенческой модели, направленная на разрешение противоречия.

Универсальные поведенческие модели обладают высокой применяемостью, однако их применение не всегда рационально. Эта модель должна обладать достаточной детализацией для применения на различных этапах проектирования СнК, что в ряде случаев неоправданно уменьшает скорость их моделирования, что является критически важным при больших

объемах задач. Таким образом, при разработке универсальных поведенческих моделей возникает техническое противоречие (рисунок 1):

- чем меньше детализация модели, тем выше скорость моделирования и меньше ее применяемость;
- чем выше детализация модели, тем ниже скорость моделирования и больше ее применяемость.

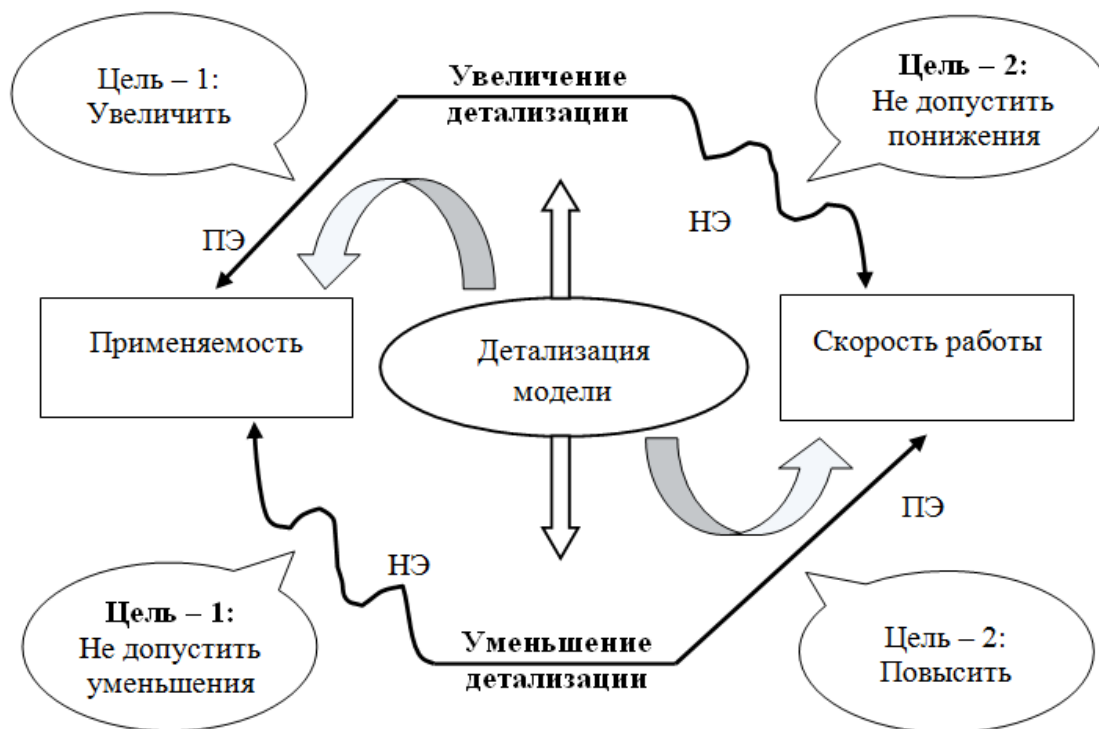


Рисунок 1 - Схема технического противоречия при проектировании универсальных поведенческих моделей (ПЭ – положительный эффект, НЭ – нежелательный эффект)

На основе анализа построенной карты противоречий было предложено отказаться от детализированного внутреннего описания модели, соблюдая при этом временные характеристики блока.

Компромиссным решением, является разработка функциональной модели без учета временных задержек с дополнительным блоком-арбитром. Арбитр распределяет эти задержки на входные и выходные сигналы, имитируя реальную работу устройства. Данная конфигурация позволит использовать разрабатываемую модель для верификации и в то же время сократить время на разработку потактовой поведенческой модели. Также предусмотрена возможность отключения блока арбитра, для того, чтобы модель можно было применять для архитектурного анализа. Поскольку разрабатываемая поведенческая модель блока инъекции маршрутизатора будет интегрирована в существующую регистровую модель маршрутизатора, необходимо обеспечить возможность подключения интерфейсов,

через которые будет происходить обмен данными между моделями. Эти интерфейсы должны быть отключаемыми и соответствовать аналогичным интерфейсам регистровой модели.

Во второй главе показаны основные принципы передачи данных в коммуникационной сети. Большое внимание уделено проблеме взаимных блокировок – ситуации, когда группа пакетов не может продолжить движение из-за ожидания освобождения ресурсов, занятых друг другом. Возникновение взаимных блокировок невозможно в суперкомпьютере «Ангара», что обеспечивается логикой его работы и принципами маршрутизации. Их исключение достигается за счет выполнения правила виртуальной сквозной маршрутизации и правила пузырьковой маршрутизации. Изложена структура пакета данных. Информация в коммуникационной сети передается по пакетам, содержащим передаваемые данные и служебную информацию, необходимую для следования пакета по сети. Пакет делится на флиты – неделимую часть информации, размер которой равен ширине физического канала. Служебных флит в пакете три: головной, служебный и конечный. Большое внимание уделено структуре головного флита, поскольку этот флит содержит в себе основную информацию, необходимую для передачи данных. Представлены основные алгоритмы, позволяющие представить работу маршрутизаторов: нахождение множества путей следования пакета по сети, выбора оптимального пути и модификация головного флита в процессе следования пакета. Изложенные алгоритмы использованы при проектировании поведенческой модели блока инъекции.

При выборе пути следования пакета данных на вход маршрутизатора поступает следующая маршрутная информация:

- координаты узла-получателя: (d_1, \dots, d_K) ;
- биты-указатели движения по направлениям (t_1, \dots, t_K) ;
- бит наличия нестандартного первого шага fs ;
- направление нестандартного последнего шага (если такой есть) ls ;
- бит разрешения адаптивности aa ;
- уровень приоритета pr .

В случае адаптивной маршрутизации также будут задействованы j_{det} и v_{det} , задающие виртуальный канал для детерминированного маршрута.

Во время работы алгоритма заголовок HF может быть изменен, о чем будет сигнализировать флаг *altered*.

Первый флит загружается в регистр HF , и сбрасывается флаг $altered$:

$$\begin{aligned} (HF) &\leftarrow (data). \\ altered &= 0 \end{aligned}$$

Координаты узла-получателя d_j соответствуют битам поля DST заголовка таким образом:

$$d_j \equiv d_j^1 d_j^2 d_j^3 d_j^4, j = 1, \dots, K;$$

t -биты соответствуют элементам поля $HF.DIR$, а биты fs , aa и pr — полям FS, A и Pr соответственно.

Маршрутная информация необходимая для передачи пакета считывается из головного флита. Соответствие полей заголовка переменным, используемым в алгоритме, показано в таблице 1.

Таблица 1 – Соответствие полей головного флита переменным алгоритма

d_j	$HF.DST[4(i-1)+1], i = (1..K)$
t_j	$HF.DIR[j], j = 1, \dots, N_{dir}$
fs	$HF.FS$
ls	$HF.LS$
aa	$HF.A$
hp	$HF.Pr$

В зависимости от приоритета будет использоваться высокоприоритетный канал или детерминированный:

$$pr = \begin{cases} 1 \Rightarrow v_{det} \leftarrow \{hp\} \\ 0 \Rightarrow v_{det} \leftarrow \{esc\} \end{cases}$$

Следующая операция несложно реализуется на логических элементах: требуется определить первый ненулевой бит среди t_j , если такой существует.

Если $\forall i, i \in (1..K), t_i = 0$ - пакет достиг узла назначения, необходимо проверить наличие нестандартного последнего шага, и, если такого нет, то есть провести эжекцию. При эжекции, в качестве выхода устанавливаем канал эжекции, маршрутизация считается детерминированной. Дополнительных условий для начала передачи не нужно, наличие места хотя бы для одного флита в буфере эжекции будет автоматически гарантировано интерфейсом входного буфера.

Если же необходимо сделать нестандартный последний шаг, необходимо определить его направление. Значение бита ls задает направление $-X_{ls}$, то е. направление номер $ls + K$. Поскольку пакет направляется в детерминированную подсеть, условие начала передачи определяется функцией BR_2 . При наличии нестандартного последнего шага адаптивная маршрутизации запрещена, и (j_{det}, v_{det}) просматриваться не будут, поэтому их устанавливать не надо.

Таким образом, представленный алгоритм можно записать в следующем виде:

$$\forall i, i \in (1..K), t_i = 0 \Rightarrow \begin{cases} ls = 000_2 \Rightarrow \begin{matrix} (j_{out}) \\ det \end{matrix} \leftarrow \begin{matrix} (0) \\ 1 \end{matrix}, br \leftarrow ready \\ ls \neq 000_2 \Rightarrow \begin{matrix} (j_{out}) \\ det \end{matrix} \leftarrow \begin{matrix} (ls + K) \\ 1 \end{matrix}, br \leftarrow BR_2 \end{cases} \quad (1)$$

Для случаев, не указанных в (1), то есть если $\exists i, i \in (1..K), t_i \neq 0$, пакет необходимо передать следующему узлу, согласно разрешенным направлениям движения. Выбираем направление, по которому предстоит двигаться в случае выбора детерминированного маршрута. В данном случае важно, что t_j упорядочены согласно правилу порядка направлений.

Далее проверяется наличие нестандартного первого шага. Если необходимо сделать нестандартный первый шаг, его направление задается в t битах. Затем определяется условие начала передачи, изменяется заголовок пакета (сбрасывается бит fs и бит соответствующего направления).

При отсутствии нестандартного первого шага в первую очередь проверяется возможность адаптивной передачи. Если соответствующей отметки в пакете нет, выбирается детерминированный канал и пакет передается. Адаптивная маршрутизация будет рассмотрена подробно в следующем разделе главы. Формализуем алгоритм передачи пакета:

$$\exists i, i \in (1..K) t_i \neq 0 \Rightarrow j_{det} \leftarrow (t_1..t_K), fs = \begin{cases} 1 \Rightarrow \begin{matrix} (j_{out}) \\ v_{out} \\ det \end{matrix} \leftarrow \begin{matrix} (j_{det}) \\ v_{det} \\ 1 \end{matrix}, \begin{matrix} (fs) \\ t_i \end{matrix} \leftarrow \begin{matrix} (0) \\ 0 \end{matrix}, br \leftarrow BR_2, altered = 1 \\ 0 \Rightarrow aa = 0, \begin{matrix} (j_{out}) \\ v_{out} \\ det \end{matrix} \leftarrow \begin{matrix} (j_{det}) \\ v_{det} \\ 1 \end{matrix}, br \leftarrow BR_1 \end{cases} \quad (2)$$

Результатом работы алгоритма являются выходной виртуальный канал, задаваемый номером направления j_{out} и типом виртуального канала v_{out} , условие начала передачи br , а также признак выбора детерминированного маршрута det .

В третьей главе изложен процесс проектирования поведенческой модели блока инжекции маршрутизатора «Ангара». Проектирование велось с использованием стандарта UML. В ходе работы были разработаны модель вариантов использования, логическая модель и модель развертывания. В рамках модели вариантов использования были разработаны диаграмма вариантов использования и диаграммы последовательности. В первом разделе главы построена диаграмма использования, отражающая функциональное назначение системы. В ходе разработки диаграммы были выявлены основные задачи, выполняемые моделью, а также были обозначены требования, предъявляемые к ее работе. Выявленные задачи были уточнены и подробно рассмотрены с использованием диаграмм последовательности. Была также представлена логическая модель. В состав разработанной логической модели вошли диаграмма пакетов, диаграмма классов и диаграмма компонентов. В ходе разработки поведенческой модели, все ее компоненты были сгруппированы в три группы. Полученная структура системы была отражена в диаграмме пакетов: каждый пакет состоит из семантически близких элементов.

Информационная система была разделена на несколько модулей, представленных в виде следующих трех пакетов: «Data», «Components», «Signals». Взаимодействие между этими пакетами представлено на рисунке 2.

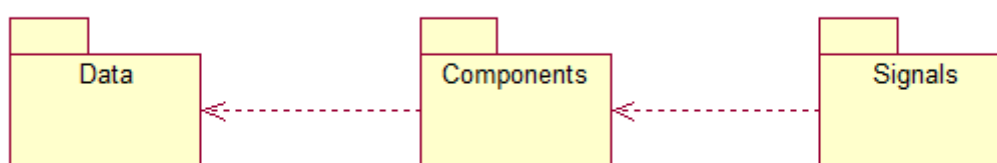


Рисунок 2 - Диаграмма пакетов поведенческой модели блока инжекции маршрутизатора

Пакеты «Data» и «Components» представляют собой набор классов реализованных на языке программирования C++. Пакет «Signals» использует библиотеку SystemC для возможности интеграции разрабатываемого программного обеспечения в регистровую модель маршрутизатора, а также для реализации параллельных вычислений при моделировании. Разработанная диаграмма пакетов позволит правильно определить

структуру программного кода, а также разделять компоненты модели в зависимости от ее применения.

Более подробно пакет «Components» рассмотрен в диаграмме классов, Показанной на рисунке 3. Полученная диаграмма представляет функционал блока инъекции, без учета интерфейсов с другими блоками.

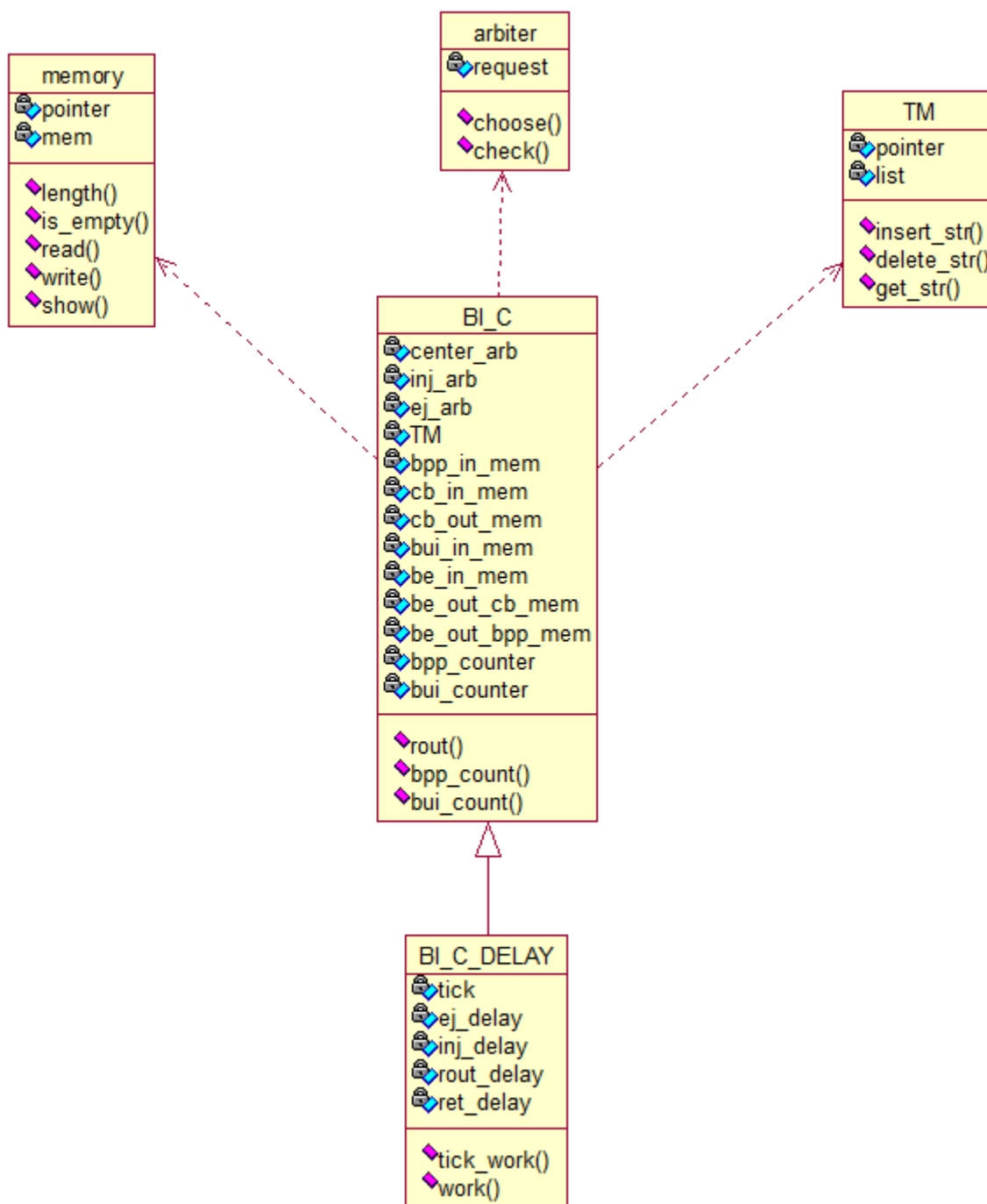


Рисунок 3 - Диаграмма классов пакета «Components»

Была также создана диаграмма компонентов, отражающая физическую структуру модели. Каждый компонент является физическим модулем программного кода.

Была разработана модель развертывания, состоящая из диаграммы развертывания. Диаграмма развертывания отражает количественный и качественный состав программно-аппаратных средств, на которых будет выполнено развертывание поведенческой модели блока инъекции. Система построена таким образом, чтобы каждый сотрудник компании смог запустить моделирование для своих нужд со своего рабочего места. Моделирование работы коммуникационной сети происходит на вычислительном сервере предприятия. Запуск моделирования осуществляется с локальных компьютеров работников с возможностью настройки параметров запускаемой системы.

Разработанные диаграммы отражают функциональные возможности создаваемого программного обеспечения и позволяют структурировать исходные данные необходимые для его создания. Проведенная работа позволит избежать ошибок проектирования системы и тем самым сократит сроки ее разработки.

В четвертой главе представлена методика тестирования разработанной поведенческой модели. Для верификации созданного программного обеспечения создается тестовое окружение – программный модуль, подающий тестовые входные воздействия на блок и контролирующий правильность выходных данных. Так же как и при разработке поведенческой модели, тестовое окружение разделяется на несколько частей. Для тестирования основного функционального компонента разработанной поведенческой модели, первую часть тестового окружения создается на языке C++. При этом, передача входных тестовых воздействий в модель осуществляется их записью во входные буферы модели. Тестирование потактовой модели с SystemC интерфейсами, осуществляется так же с использованием данной библиотеки. В данном случае, передача входных тестовых данных происходит через SystemC сигналы. Алгоритм и структура тестового окружения для представленных тестов одинаков. Так как интерфейсы и функциональность потактовой поведенческой SystemC модели в точности совпадают с регистровой, в дальнейшем разрабатываемое тестовое окружение может быть применено и для верификации регистровой модели путем простой замены одной модели на другую.

После проведения тестирования разработанной поведенческой модели была проведена оценка скорости моделирования регистровой модели с использованием уже разработанного тестового окружения. Поскольку во время моделирования регистровой и поведенческой модели тестовое окружение не менялось, его влияние на скорость моделирования одинаково и, таким образом, результаты тестов можно считать сравнимыми.

Полученные результаты работы двух моделей были сопоставлены между собой. Результаты проведенных исследований представлены в таблице 2.

Таблица 2 – скорости моделирования поведенческой и регистровых моделей в зависимости от загруженности интерфейсов

Интерфейс Модель	БЭ	БУИ	БПП	Коммутатор	Все интерфейсы
RTL	1300	900	500	1100	300
Поведенческая	16500	8100	4200	9600	3000

Сравнение скорости моделирования работы поведенческой и регистровой моделей при различной загруженности интерфейсов, показало, что тактовая частота работы виртуального кристалла при работе поведенческой модели в среднем в 10 раз выше данного показателя при работе регистровой модели. Такое же отношение скоростей моделирования было получено при максимальной загруженности модели.

Полученные результаты говорят о том, что поставленная задача повышения скорости моделирования, была выполнена. Проведение тестирования отдельных участков кода модели не выявило значительных задержек работы, в каких либо отдельных модулях поведенческой модели блока инъекции.

Для оценки принятых решений по архитектуре маршрутизатора и алгоритмов его работы применяется моделирование коммуникационной сети. Результаты моделирования дают разработчикам наглядное представление о работе проектируемой сети, позволяют зафиксировать задержки при передаче данных, а также выявить «узкие места» разрабатываемой системы. Использование регистровых моделей позволяет оценить правильность работы разработанного кода.

Основной проблемой существующей модели является низкая скорость моделирования работы сети: работа одной секунды сети из 9 узлов моделируется в течение месяца. Данный факт приводит к тому, что разработчики не имеют возможности оценить изменения, внесенные в структуру маршрутизатора при его работе в сети в течение продолжительного количества времени. По той же причине данная модель не используется для отладки программного обеспечения.

Увеличение скорости моделирования сети может быть решено использованием поведенческих моделей. Повсеместная замена регистровых моделей поведенческими,

увеличит скорость моделирования в десятки раз. При этом с целью верификации в одном из узлов будет оставлена регистровая модель.

Модель маршрутизатора собирается из соответствующих моделей его логических блоков. По мере их разработки, они будут заменять собой регистровые модели. Такая структура увеличит взаимозаменяемость и позволит верифицировать не только весь маршрутизатор в целом, но и отдельные блоки без использования дополнительного тестового окружения.

Поскольку интерфейсы поведенческой модели в точности повторяют интерфейсы регистровой модели, замена моделей была произведена без каких либо изменений в их коде. Для анализа скорости моделирования работы коммуникационной сети фиксировались промежутки времени между началом инъекции пакетов и окончанием моделирования, и рассчитывалась тактовая частота виртуального кристалла. Диаграмма скоростей моделирования коммуникационной сети с использованием поведенческой модели и без нее показана на рисунке 5.

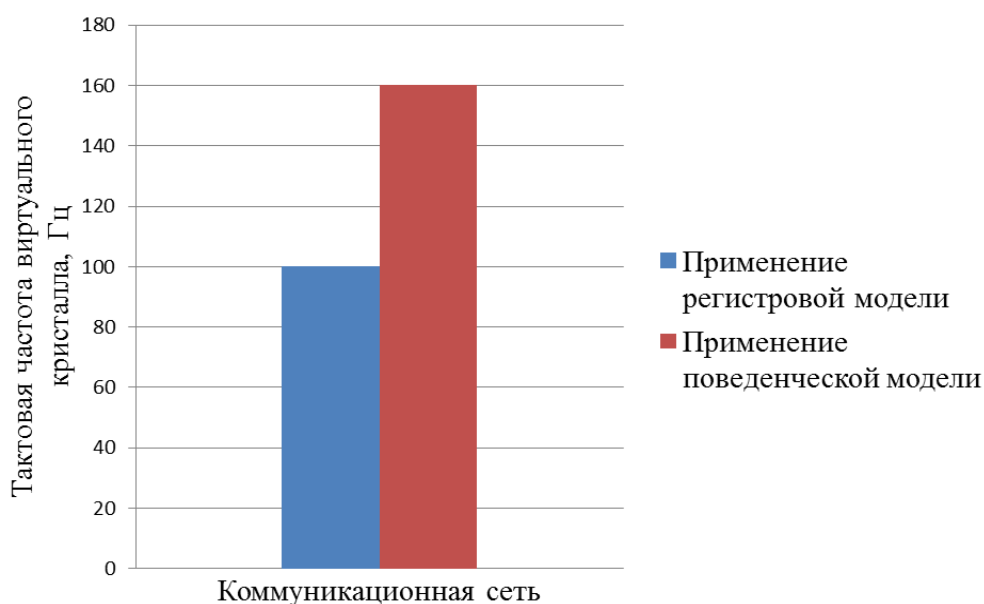


Рисунок 5 – Столбиковая диаграмма скорости моделирования маршрутизатора с использованием разных моделей

Моделирование сети с различными моделями производилось при прочих равных условиях: тестовое окружение, методика тестирования не изменялись. Как видно из рисунка 5 замена регистровой модели на поведенческую увеличило скорость моделирования в 1,6 раза: тактовая частота виртуального кристалла выросла в среднем со 100 Гц до 160Гц.

Исходя из того факта, что в блоке инъекции маршрутизатора, выполняется 12% операций от всего устройства, можно предположить, что при замене всех моделей блоков маршрутизатора частота виртуального кристалла вырастет в 13 раз.

В заключении подводится итог выполненной работы. Изложены результаты проектирования поведенческой модели блока инъекции маршрутизатора «Ангара». Приведены результаты проведения моделирования разработанной модели и дана оценка ее применимости в рамках маршрута проектирования систем на кристалле. Указаны сферы науки и техники, которым могут быть полезны результаты, полученные в данной квалификационной работе.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Тактовая частота работы виртуального кристалла при моделировании работы блока инъекции маршрутизатора «Ангара» при использовании разработанной программной поведенческой модели составила 3000 Гц, что в 10 раз превышает данный показатель работы RTL кода.
2. Разработанная программная поведенческая модель была интегрирована в модель коммуникационной сети «Ангара» Использование разработанной модели позволило увеличить тактовую частоту работы виртуального кристалла при моделировании работы сети в 1,6 раз. На основании достигнутых результатов был сделан вывод о целесообразности проводимых разработок и принято решение о продолжении работы в данном направлении.

ОСНОВНЫЕ ПОЛОЖЕНИЯ РАБОТЫ ИЗЛОЖЕНЫ В СЛЕДУЮЩИХ РАБОТАХ:

1. Котельницкий А.В. Использование программных поведенческих моделей при проектировании маршрутизатора суперкомпьютерной сети «Ангара» // Сборник докладов 14 - молодежной международной научно-технической конференции «Наукоемкие технологии и интеллектуальные системы» - Москва, МГТУ им.Н.Э.Баумана, 25 апреля 2012 г. - С.312-315.
2. Власов А. И., Котельницкий А.В. Применение поведенческих моделей при проектировании систем на кристалле // Наука и образование: электронное научно-техническое издание. [в печати].