электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 00

Верификация коммуникационной сети топологии «4-d тор»

 # 09, сентябрь 2012

Иванов А. М.

Научный руководитель: доцент, к.т.н., Власов А.И.
МГТУ им. Баумана, Москва, РФ

МГТУ им. Н.Э. Баумана
bauman@bmstu.ru

Введение

На сегодняшний день в области создания распределенных систем передачи и обработки данных сложилась диспропорция темпов развития трёх её основных компонентов: микропроцессорной техники, средств телекоммуникации и разработки программного обеспечения [1]. Это приводит к тому, что разрыв между техническими возможностями и технологиями обработки и передачи информации всё время увеличивается. В значительной степени, это происходит из-за отсутствия удовлетворительного для промышленности решения проблемы проверки правильности программных систем и логических схем вычислительных устройств. Например, из практики разработки программного обеспечения хорошо известно, что примерно 2/3 времени, затрачиваемого на создание системы, приходится на отладку, т.е. проверку её правильности. Как правило, большинством разработчиков программных систем для проверки правильности проекта практикуются **методы имитационного моделирования и тестирования** [1]. Они довольно эффективны на самых ранних стадиях отладки, когда проектируемая система всё ещё изобилует ошибками, но результативность этих методов быстро снижается, как только система становится чище. Достойной альтернативой имитационному моделированию и тестированию являются методы **формальной верификации**. При имитационном моделировании и тестировании исследуются только некоторые из возможных сценариев поведения проектируемой системы, поэтому остаётся открытым вопрос о том, не содержится ли фатальная ошибка в незадействованных траекториях. Формальная верификация же обеспечивает исчерпывающий анализ всех возможных вариантов поведения системы. Техника верификации, получившая название **проверки на модели**, является одним из наиболее перспективных и широко используемых подходов к решению проблемы автоматизации отладки и проверки правильности программ. Данный метод применялся и при создании тестового окружения для коммуникационной сети топологии «4D Тор» в компании «НИЦЭВТ». По сравнению с другими подходами в формальной верификации про грамм, метод проверки на модели обладает двумя замечательными преимуществами [1].

Он полностью автоматический, и его применение не требует от пользователя никаких особых знаний в таких математических дисциплинах, как логика и теория доказательства теорем. Всякий, кто может про вести моделирование проектируемой системы, вполне способен осуществить и проверку этой системы.

Если проектируемая система не обладает желаемым свойством, то результатом проверки на модели будет контрпример, который демонстрирует поведение системы,

опровергающее это свойство. Эта ошибочная трасса даёт бесценную информацию для понимания причины ошибки, и важный ключ к решению возникшей проблемы.

1 Принципы построения тестового окружения

Кроме внешнего окружения, в рамках которого должна работать среда верификации, необходимо также определить ее **каркас** — некоторый базовый набор компонентов, реализующих основной набор функций и поддерживающие основные потоки данных внутри системы. К этому каркасу будут добавляться другие компоненты, поддерживающие вспомогательные и менее значимые функции [2]. Предлагается использовать в качестве основы для построения среды верификации архитектурный каркас ***RTL*-модели** маршрутизатора. Это решение вызвано тем обстоятельством, что на построение собственной модели для тестирования тратится неоправданно много времени. Вместо полноценной тестовой модели создается надстройка над *RTL*-модулем в виде интерфейса с высокоуровневым тестовым окружением.

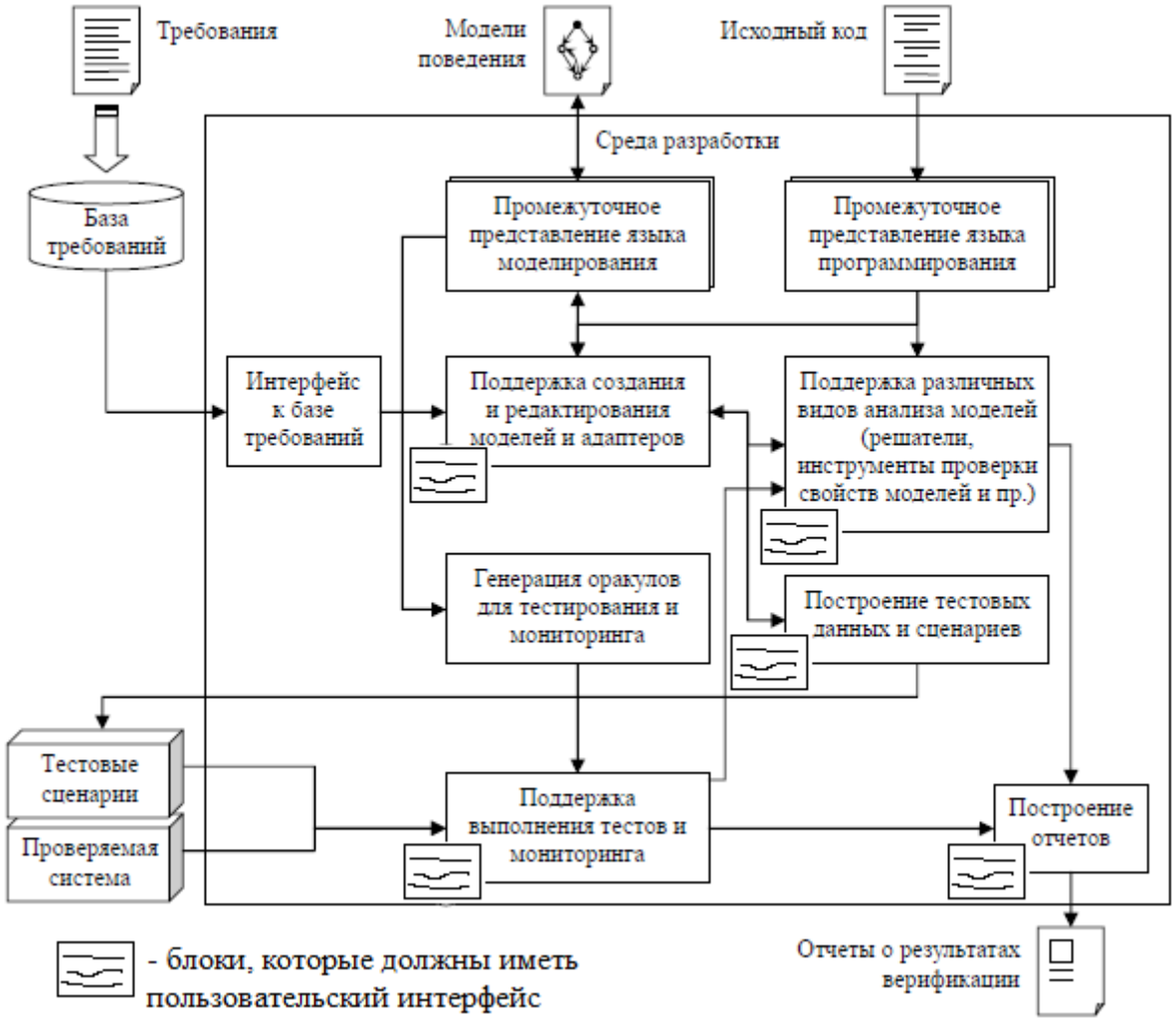


Рис. 1 Архитектура системы верификации программного обеспечения

Обычно в ходе тестирования на основе моделей необходимо сделать следующее [2]:

- определить модель поведения тестируемой системы, формализующую требования к этому поведению.
- проанализировать структуру модели для выбора критериев покрытия и отдельных целей тестирования, и определить эти критерии и цели.

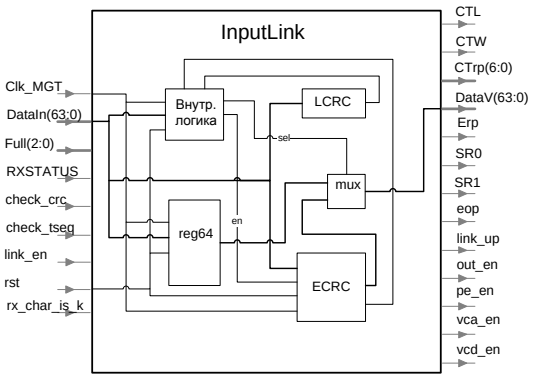
- построить среду выполнения тестов, включающую средства мониторинга для протоколирования внешних действий, реакций системы, и, возможно, внутренних ее событий, а также тестовые оракулы (рис.1) — программные компоненты, определяющие соответствие или несоответствие наблюдаемого поведения системы и модели. Обычно такая среда состоит из библиотеки поддержки выполнения тестов, набора тестовых оракулов для всех проверяемых компонентов и набора адаптеров, связывающих эти компоненты с их оракулами. Оракулы в большинстве случаев генерируются автоматически из ранее построенной модели.
- построить, автоматически или с привлечением человека, набор тестовых сценариев, определяющих последовательности вызова различных операций тестируемой системы или посылки ей сообщений или сигналов и данные, передаваемые в качестве параметров операций и сообщений.
- выполнить тестовые сценарии, протоколируя всю информацию, касающуюся соответствия наблюдаемого поведения системы и ее модели, а также покрытых во время тестирования ситуаций.
- провести анализ результатов тестов, в ходе которого выявляются и анализируются ошибки в системе или ее модели (проявляющиеся как несоответствия между ожидаемым и реальным поведением), а также анализируется достигнутое тестовое покрытие и принимается решение либо о создании дополнительных тестов, либо об окончании их разработки [3].

2 Верификация блока «Link» маршрутизатора

Блок «*Link*» маршрутизатора состоит из двух частей. Блок анализа и демультиплексор (БАД) предназначен для приема информационных пакетов из сети и направлении их по одному из виртуальных каналов в соответствии с маршрутом. Блок передачи данных (БПД) осуществляет справедливый арбитраж пакетов (пропускание пакетов в сеть из нескольких виртуальных каналов последовательно в соответствии с очередностью). Так же данные блоки должны удовлетворять требованиям быстродействия (обеспечивать минимальные задержки прохождения пакетов) и отказоустойчивости (обеспечивать гарантии сохранения целостности пакета при прохождении его через сеть). Целью построения тестового окружения является:

- проверка правильности приема пакетов;
- оценка быстродействия прохождения пакетов через *Link;*
- проверка арбитража пакетов;
- оценка отказоустойчивости блока.

Внутреннее устройство данных блоков и их интерфейсы с другими модулями маршрутизатора показаны на рисунках 2 и 3.



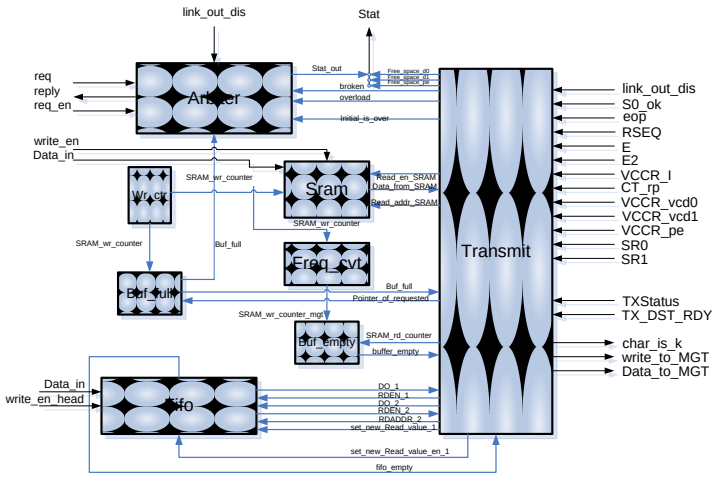


Рис. 2 Блок анализа и демультиплексор Рис. 3 Блок передачи данных

В проектирования создания системы верификации была создана надстройка для связи RTL-модели Линка с внешним тестовым окружением. Так же создан **драйвер** – генератор входных сигналов, который определяет воздействия на RTL, различные счетчики и события для реализации тестовых сценариев, **монитор –** интерфейс пользователя, который отображает факты прохождения тестов и статистические данные.

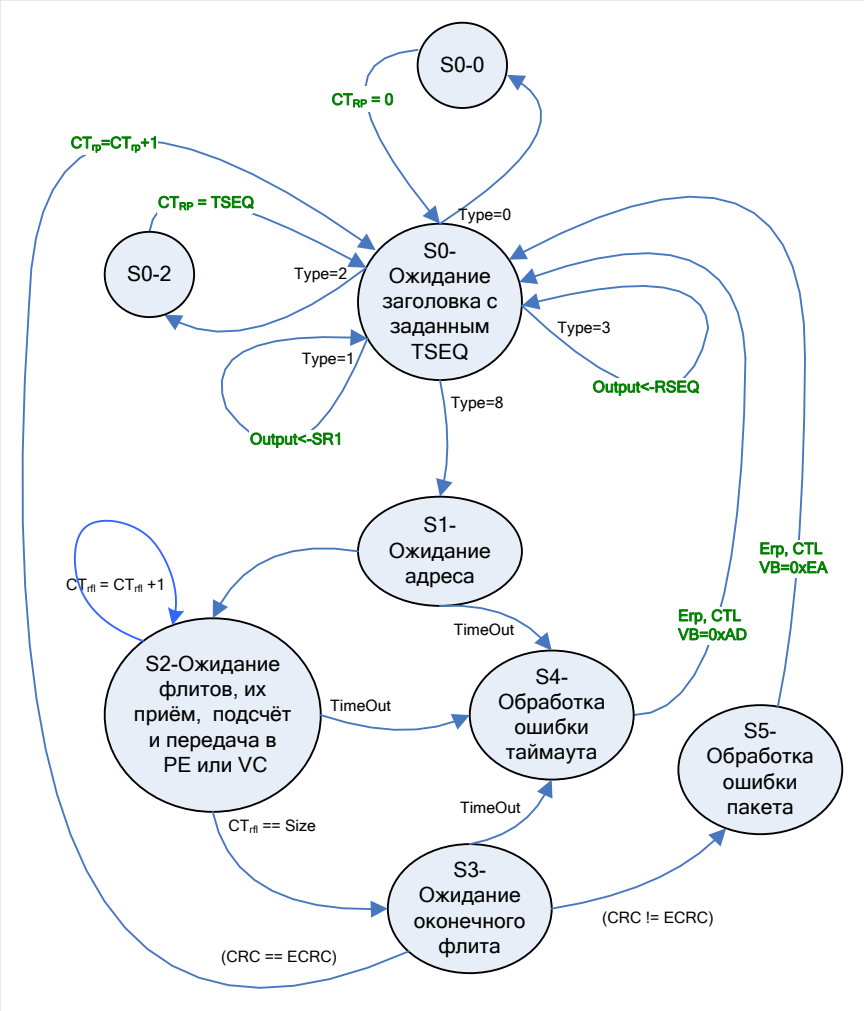


Рис. 4 Диаграмма состояний блока БАД

В результате тестирования проверялись следующие функции Линка.

Для блока БАД:

- приём пакетов из сети;
- анализ пакетов и передача в соответствующий виртуальный канал или входную очередь локального вычислительного узла;
- контроль целостности данных пакета;
- определение направления передачи пакетов на основании номера виртуального канала;
- контроль целостности данных путём подсчёта пакетов, контроля CRC (контрольной суммы) заголовочного флита и тела пакета;

Тестовые сценарии для блока БАД строились на основании диаграммы состояний, показанной на рисунке 4.

Для блока БПД:

- способность осуществлять справедливый арбитраж запросов и установление соединений посредством кроссбара с различными виртуальными каналами маршрутизатора;

- прием данных с выхода кроссбара и помещение их во внутренний буфер БПД;

- передача принятых во внутренний буфер БПД данных, не дожидаясь завершения приёма, на следующий маршрутизатор;

- осуществление контроля и удаления из внутреннего буфера БПД пакетов, успешно принятых следующим маршрутизатором, а также повторную передачу пакетов при возникновении ошибки на приёмной стороне).

Исследования работы Линка проводились с использованием языка моделирования и верификации SystemC, а так же с применением среды разработки тестового окружения ModelSim фирмы «Mentor Graphics» [3].

Литература

1. Гаранина Н.О., «Верификация распределенных систем с использованием аффинного представления данных, логик знаний и действий» Дис. канд. физ.-мат. Наук, Новосибирск, 2004.
2. Кулямин В.В.: «Интеграция методов верификации программных систем», ИСП РАН, Москва, 2008.
3. А.Лохов «Функциональная верификация СБИС в свете решений Mentor Graphics», выст. «Метрология и измерительные системы» 2004.