

ПРОГРАММНАЯ СИСТЕМА ДЛЯ ИССЛЕДОВАНИЯ ПРОТОКОЛОВ СИНХРОНИЗАЦИИ ПРИ РАСПРЕДЕЛЕННОМ СОБЫТИЙНОМ ЛОГИЧЕСКОМ МОДЕЛИРОВАНИИ

Ладыженский Ю.В., Попов Ю.В.

Донецкий национальный технический университет, кафедра ПМИИ

E-mail: ly@cs.dgtu.donetsk.ua

Abstract

Ladyzhensky Y.V., Popoff Y.V. A system for research of synchronization protocols for distributed event-driven logical simulation. Analysis of synchronization protocols for logic simulations is fulfilled. Algorithms for experimental research of parallel simulations are developed. A system architecture is proposed. Software was implemented and tested in a local network.

Введение

Моделирование поведения во времени логических схем является важным этапом создания цифровых устройств. Высокая размерность и сложность реальных логических схем приводит к большим затратам времени и памяти при моделировании. Ускорение моделирования и ресурсы памяти можно получить при использовании нескольких процессоров, работающих параллельно. Для получения наилучших результатов при распределенном моделировании дискретных логических схем необходимо исследовать работу различных протоколов синхронизации логических процессов. Для исследования необходима программная система, выполняющая моделирование, сбор информации о ходе моделирования и анализ полученных данных.

Анализ последних исследований показал, что использование различных протоколов синхронизации логических процессов при распределенном логическом моделировании оказывает существенное влияние на эффективность моделирования. В [1] представлены методы исследования цифровых систем с программируемой логикой средствами моделирования. В [2] рассмотрены средства разработки цифровых устройств. Важным моментом является то, что в этой статье рассматриваются проблемы обучения технологиям улучшения средств разработки цифровых устройств в технических университетах. В [3] рассматривается реализация системы распределенного логического моделирования на основе консервативного протокола синхронизации логических процессов; предложена архитектура системы и приведены алгоритмы функционирования основных подсистем. Система для исследования методов параллельного логического моделирования рассмотрена в [4]. Основные алгоритмы, используемые при разработке многопоточных и параллельных, параллельных программных систем, рассмотрены в [5].

Цели и задачи исследований, рассмотренных в статье, включают в себя разработку архитектуры системы для экспериментального исследования различных протоколов синхронизации логических процессов при распределенном моделировании. Проведение объектно-ориентированного анализа архитектуры консервативного и оптимистического процессов моделирования. Выделение объектов, необходимых для выполнения большинства протоколов синхронизации логических процессов. Разработка средств анализа результатов распределенного логического моделирования, которые должны обеспечивать построение диаграмм причинно-следственных связей (ПСС-диаграмм), построение временных диаграмм и обработку статистической информации.

Архитектура программной системы

Для исследования работы различных протоколов синхронизации (ПС) логических процессов (ЛП) при распределенном логическом моделировании (РЛМ) необходима специальная программная система. Разработанная программная система для исследования ПС ЛП при РЛМ включает в себя подсистему ввода исходных схем и входного воздействия, подсистему РЛМ и подсистему анализа результатов моделирования. Архитектура программной системы приведена на рис. 1.

Разработанная программная система предназначена для:

- ввода исходных схем, представленных многоуровневым описанием, ввода входного воздействия, автоматического генерирования случайных схем и входных воздействий с заданными параметрами, автоматического разрезания исходной схемы для моделирования на нескольких процессорах с заданными критериями оптимальности разрезания;
- проведения распределенного моделирования с использованием выбранного протокола синхронизации, автоматического распределения задач моделирования по свободным процессорам, автоматического сбора информации о ходе моделирования;
- упрощения задачи программирования новых протоколов синхронизации ЛП путем предоставления доступа к объектам, которые используются большинством протоколов синхронизации при РЛМ;

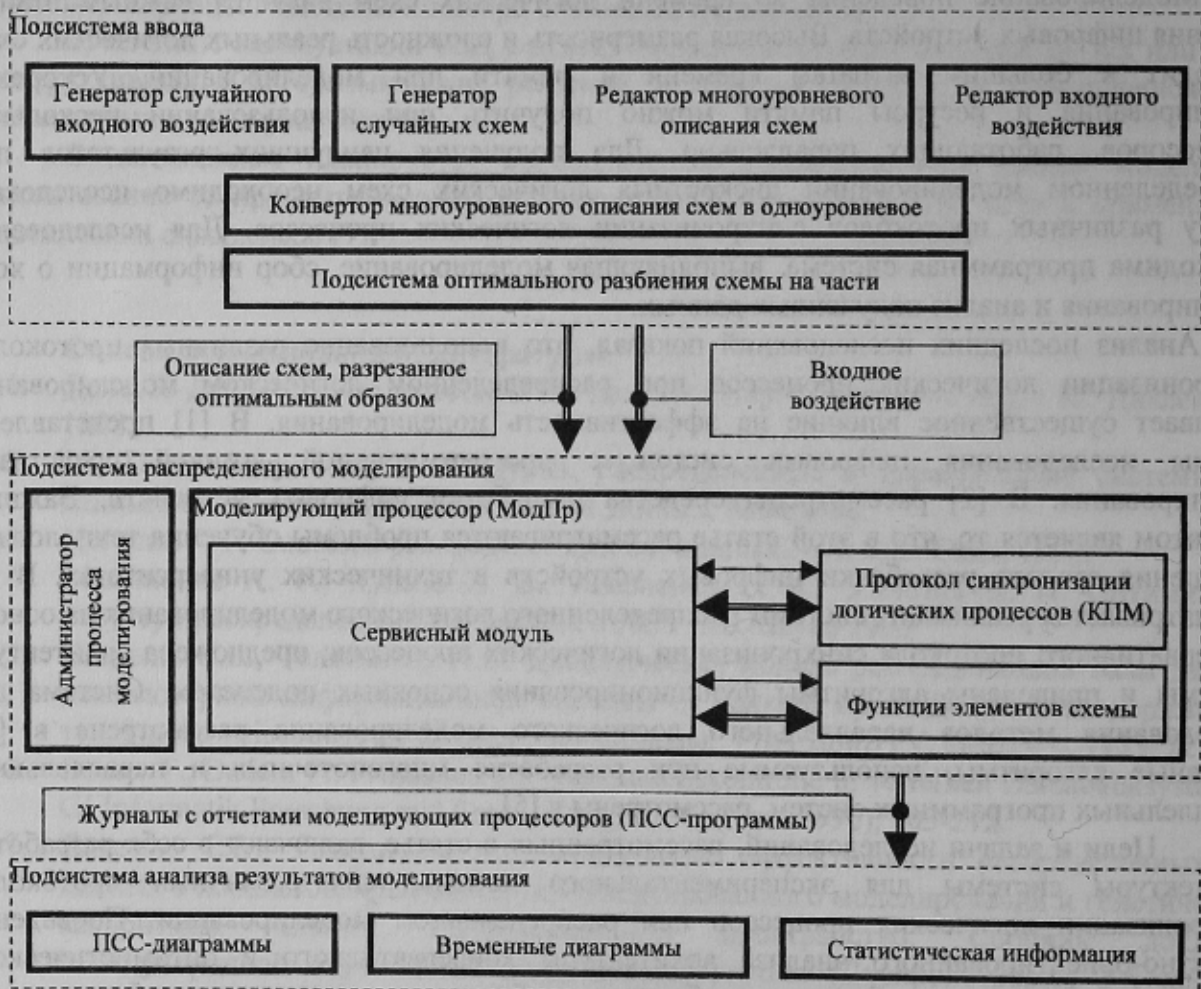


Рисунок 1 - Архитектура программной системы для исследования протоколов синхронизации логических процессов при распределенном моделировании

- анализа результатов моделирования. Для этого программная система включает в себя компоненты для построения диаграмм причинно-следственных связей,

временных диаграмм и обработки статистической информации. Результаты моделирования представляются в виде программы на языке диаграмм причинно-следственных связей (ПСС-программы).

Подсистема ввода исходных данных включает в себя компоненты для случайной генерации схем и входного воздействия. Эти компоненты необходимы для того, что бы исследовать поведение протоколов синхронизации при моделировании схем с заданными параметрами. Параметры включают в себя типы используемых элементов, законы распределения типов элементов, описание графа процессоров, распределение элементов по процессорам, способ соединения элементов на процессоре: по уровням, без циклов либо случайным образом, с циклами. Компонент для оптимального разбиения схемы на части предназначен для исследования поведения протоколов синхронизации при моделировании реальных схем, которые были разрезаны согласно некоторым критериям оптимальности. Критерии оптимальности включают в себя дисбаланс нагрузки на процессоры, количество межпроцессорных соединений, количество и глубина циклов в графе процессоров.

Подсистема распределенного моделирования включает в себя один администратор процесса моделирования (АПМ) и множество моделирующих процессоров.

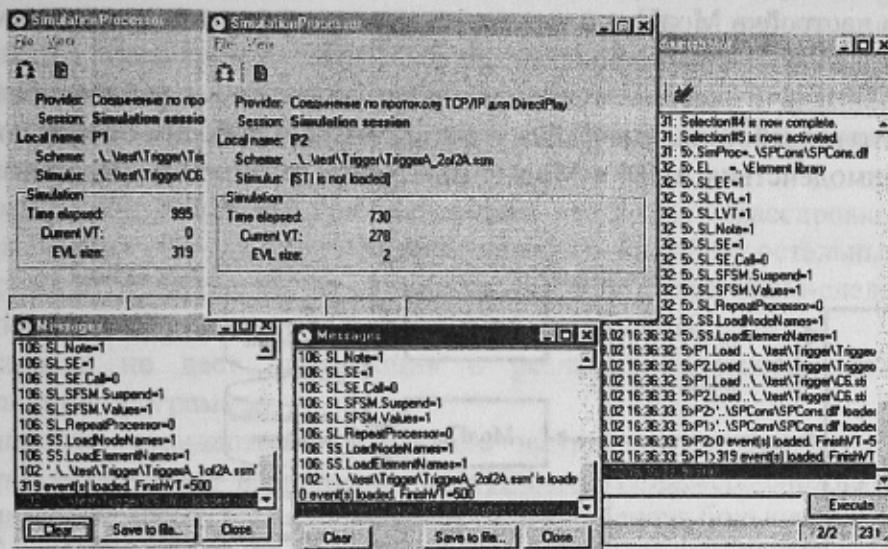


Рисунок 2 – Программная система для исследования протоколов синхронизации логических процессов при распределенном моделировании

Моделирующий процессор состоит из трех модулей.

Сервисный модуль содержит в себе реализацию основных объектов, которые необходимы при использовании большинства протоколов синхронизации.

ПС ЛП содержит в себе координатор процесса моделирования (КПМ) и специфические для данного ПС объекты. КПМ предназначен для обработки событий. Специфические объекты, в зависимости от выбранного ПС, могут включать в себя подсистему определения/восстановления для тупиков при консервативном моделировании или подсистему управления состояниями и определения глобальных состояний при оптимистическом моделировании.

Функции элементов схемы реализованы в виде динамически подключаемых библиотек. Функции элементов содержат в себе алгоритмы функционирования элементов, применяемых в моделируемых схемах.

Внешний вид АПМ и МодПр приведен на рис. 2. Основные функции меню – установить параметры подключения к сети и показать окно с сообщениями. Сообщения включают в себя информацию о загруженных схемах, входных воздействиях и о параметрах моделирования. На главной форме МодПр приводится общее реальное время, которое

прошло с момента начала решения задачи моделирования, текущее виртуальное время и текущий размер локального списка событий.

Подсистема распределенного моделирования:

- получает на вход описание схемы, разрезанное для моделирования на нескольких процессорах оптимальным образом и входное воздействие;
- производит распределенное моделирование схемы;
- результаты моделирования сохраняются в журналах отчетов моделирующих процессоров.

Задача моделирования одной схемы с заданным входным воздействием и заданными настройками МодПр считается единой задачей и не может быть разбита на подзадачи. Проект содержит в себе несколько задач. Если количество доступных МодПр больше, чем необходимо для решения одной из задач, то оставшиеся МодПр будут загружены решением других задач. АПМ предназначен для управления процессом решения задач по моделированию. Управление производится командами проекта АПМ. В файле проекта АПМ описываются следующие разделы:

- можно использовать константы для задания значений, которые может понадобиться изменить одновременно в нескольких местах;
- задаются настройки МодПр по умолчанию для моделирования всех схем и для каждой схемы отдельно. Эти настройки зависят от МодПр;
- для каждой задачи указываются имена процессоров, имена файлов с описанием схемы и входного воздействия, имя файла, в который будут собраны отчеты со всех МодПр.

Схема взаимодействия АПМ и МодПр при распределенном моделировании приведена на рис. 3.

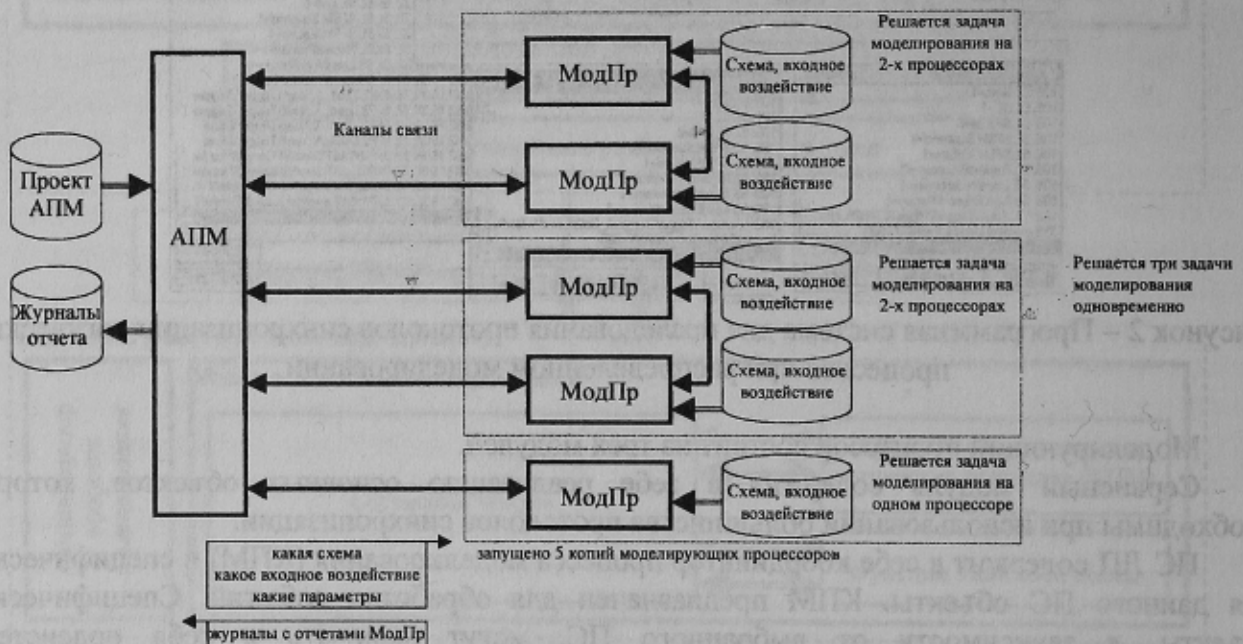


Рисунок 3 - Схема взаимодействия администратора процесса моделирования и моделирующих процессоров при распределенном моделировании

АПМ используется для решения следующих основных задач:

- из всех МодПр, подключенных к сети, выбор тех, на которых будет производиться моделирование схемы;
- установка параметров МодПр и журнала отчета перед моделированием каждой конкретной схемы;
- указание МодПр, какую схему и какое входное воздействие нужно загрузить;
- автоматический сбор результатов моделирования со всех МодПр в один файл;

- ускорение моделирования нескольких схем за счет оптимального использования процессоров. Оптимизация состоит в том, что при наличии свободных процессоров может производиться моделирование нескольких независимых схем одновременно.

Моделирующие процессоры управляются администратором процесса моделирования. АПМ отправляет по сети сообщения моделирующим процессорам, в которых содержится информация о том, какие схемы и какие входные воздействия нужно загружать, какой протокол синхронизации использовать, какие данные заносить в журнал отчета моделирования. АПМ только передает пути к файлам, а сами файлы должны быть доступны локально для каждого МодПр.

Во время моделирования каждый МодПр ведет локальный журнал отчета. В этом журнале собирается информация о процессе моделирования. По завершению моделирования отчеты моделирующих процессоров могут быть автоматически собраны АПМ в один файл.

Журнал отчета моделирующего процессора

Для анализа работы любой программной системы необходима информация о всех состояниях системы и обо всех событиях, которые в этой системе произошли. Задача анализа усложняется тем, что требуется анализировать работу распределенного приложения.

Одним из самых распространенных методов анализа работы последовательных программ является трассировка. Во время трассировки программа выполняется шаг за шагом, с остановками после каждого шага для анализа текущего состояния. Применение трассировки для анализа системы распределенного логического моделирования осложняется следующими факторами:

- задержки, возникающие при остановке во время трассировки одной копии программы могут существенным образом влиять на работу остальных копий;
- применение трассировки не позволяет выявлять причинно-следственные связи, возникающие при работе распределенного приложения.
- трассировка не дает информации о реальных задержках, возникающих при выполнении программы;
- большинство стандартных отладчиков не позволяют получить статистическую информацию о ходе выполнения программы. Например, сложно таким способом определить, какая из точек программы была выполнена большее число раз;

Для того, что бы решить перечисленные выше задачи, в каждой копии программы во время моделирования записывается последовательность состояний в файл. Такой файл называется журналом отчета моделирующего процессора. Сформулируем основные требования к журналу отчета:

- задержки, возникающие в результате использования журнала отчета, должны быть минимальными. Для этого журнал записывается в локальный файл в потоковом режиме и собирается в один файл только после завершения моделирования;
- журнал должен давать информацию о реальных задержках, возникающих во время работы программы. Для этого к каждой записи журнала добавляется реальная временная метка;
- формат журнала легко расширяем для исследования новых или модифицированных протоколов синхронизации логических процессов.

Для минимизации информации в журнал отчета записываются только события, происходящие во время моделирования. Не будет вноситься информация о переменных состояниях, которые не изменились. Во время моделирования могут возникнуть следующие основные типы событий:

- возникло событие в моделируемой схеме (внутреннее или внешнее);
- отправлено или получено сообщение по сети;
- произошла ошибка;

- изменилось состояние системы;
- пройдена контрольная точка программы.

Диаграмма классов

Эффективным средством разработки концептуальных основ проектирования систем распределенного логического моделирования является объектно-ориентированный подход. Анализ способов организации логического моделирования позволяет выделить объекты и методы их взаимодействия. К важнейшим объектам относятся: структурно-функциональная модель цифрового устройства, моделирующие процессоры, протоколы управления модельным временем, протоколы межпроцессорного обмена [4].

Диаграмма классов подсистемы распределенного логического моделирования приведена на рис. 3.

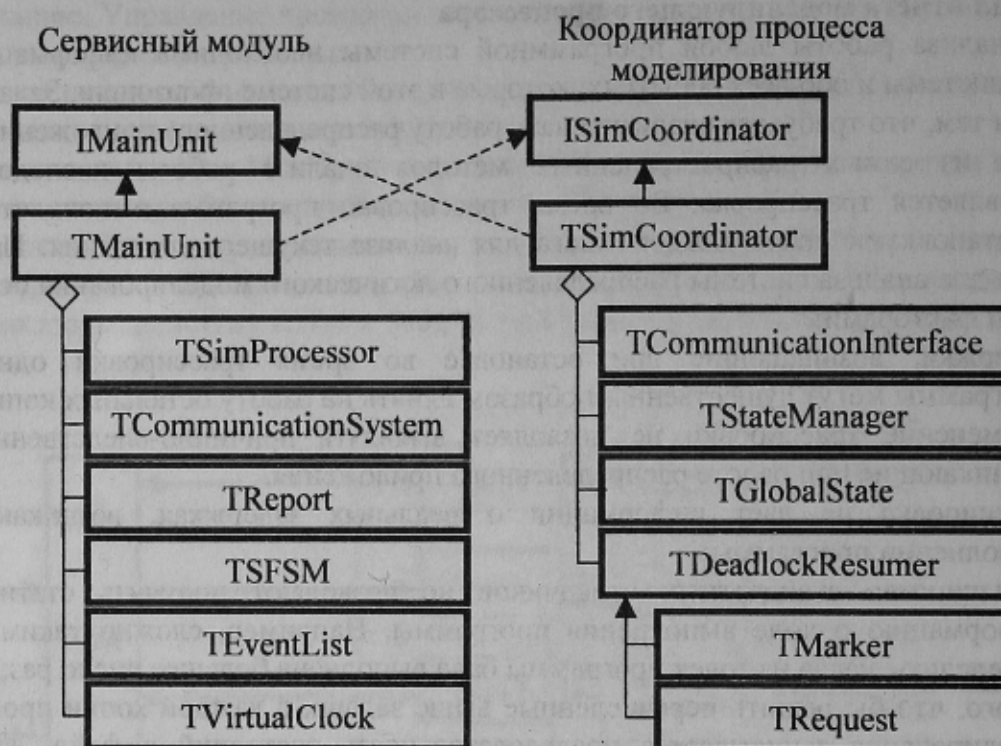


Рисунок 4 – Диаграмма классов программной системы

Сервисный модуль содержит в себе реализации объектов, необходимых при использовании большинства протоколов синхронизации ЛП. Доступ к этим объектам осуществляется через интерфейс IMainUnit. Этот интерфейс реализуется объектом класса TMainUnit. Основные объекты, к которым предоставляет доступ сервисный модуль:

- TSimProcessor – объект для управления КПМ. Позволяет загрузить нужный КПМ и запустить цикл обработки событий отдельным потоком. Отдельный поток необходим для того, что бы во время обработки событий можно было получать сообщения, приходящие из сети;
- TCommunicationSystem – объект для установки соединения с другими МодПр по сети. Является надстройкой над системой коммуникаций DirectPlay и позволяет АПМ задавать область видимости других процессоров, осуществлять доступ к процессорам по их именам и выбирать один из процессоров в качестве ведущего. Это может потребоваться, например, для запуска маркера в начале моделирования;
- TReport – объект для создания отчетов МодПр (генерации ПСС-программ);

- TSFSM – структурно-функціональна модель схеми. Содержит в себе загрузчик схем из файлов, загрузчик функций элементов, списки элементов, списки соединений и списки значений сигналов в узлах схемы;
- TEventList – список событий, которые должны произойти на данном процессоре. Содержит в себе специальную структуру данных, которая позволяет быстро добавлять новые события в любое место списка, быстро находить место в списке, куда это событие нужно вставить и быстро удалять обработанные события;
- TVirtualClock – виртуальные часы, показывают локальное виртуальное время.

Поскольку обработка событий производится отдельным потоком, все объекты, доступ к которым предоставляется сервисным модулем, содержат в себе средства синхронизации потоков.

Координатор процесса моделирования содержит в себе реализацию протокола синхронизации ЛП. Доступ к объектам КПМ осуществляется через интерфейс ISimCoordinator. Этот интерфейс реализуется объектом класса TSimCoordinator. В зависимости от используемого протокола синхронизации, КПМ может содержать в себе реализации следующих объектов:

- TCommunicationInterface – коммуникационный интерфейс. Содержится в большинстве протоколов синхронизации, но для каждого – своя реализация. Предназначен для синхронизации обработки событий на локальном процессоре с удаленными процессорами;
- TStateManager – менеджер состояний. Предназначен для предоставления возможности возврата в одно из прошлых состояний, когда из сети пришло запоздавшее событие. Содержится в оптимистических протоколах синхронизации. Организован на основе структур, подобных списку событий: можно быстро вставлять/удалять элементы и быстро производить поиск нужного места в списке;
- TGlobalState – подсистема сбора глобальных состояний. Используется в основном в оптимистических и смешанных протоколах синхронизации ЛП для определения глобального виртуального времени. Работает по алгоритму, описанному в [5];
- TDeadlockResumer – подсистема определения / восстановления системы из тупиков. Используются в основном в консервативных, но могут использоваться и в смешанных протоколах синхронизации ЛП. На данный момент реализовано два алгоритма управления тупиками: с использованием маркера и алгоритм, срабатывающий по требованию.

Диаграммы причинно-следственных связей

Во время моделирования МодПр записывают в файл информацию обо всех произошедших событиях. Запись производится на языке диаграмм причинно-следственных связей. Файл специального формата, содержащий в себе списки всех событий на всех процессорах, называется ПСС-программой. Во время моделирования каждый МодПр генерирует свой раздел ПСС-программы. После завершения все разделы записываются в один файл. Результатом интерпретации ПСС-программы является ПСС-диаграмма. Вид окна интерпретатора ПСС-программ и пример ПСС-диаграммы приведен на рис. 4.

ПСС-диаграммы предназначены для визуального представления результатов моделирования и графов причинно-следственных связей событий. ПСС-диаграммы предоставляют информацию о том, где, когда и какие события произошли, какие изменения в состоянии МодПр в связи с этими событиями имели место в удобном для восприятия виде. Анализ ПСС-диаграмм при последовательном и распределенном моделировании позволяет оценить адекватность моделирования, отсутствие ненужных задержек при моделировании и лишние сообщения в сети, определить наиболее времяемкие места в протоколе синхронизации логических процессов.

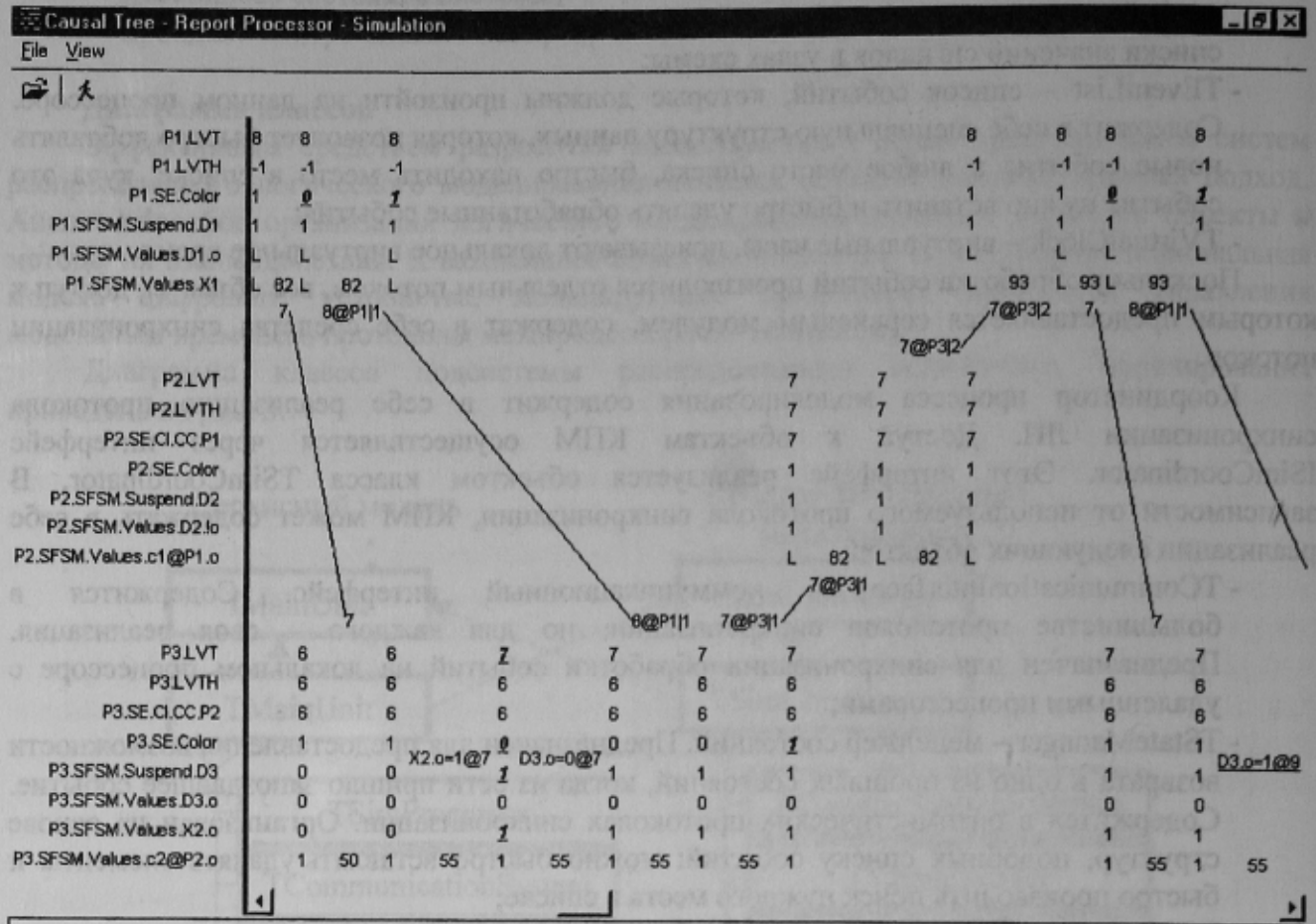


Рисунок 5 – Интерпретатор ПСС-программ

На ПСС-диаграммах представлены обработанные события, отправленные и полученные сообщения в сети, значения сигналов в узлах моделируемой схемы, значения виртуального и реального времени для каждого события, списки необработанных элементов. Для каждого события приводится полная информация о состоянии МодПр, на котором это событие произошло. Параллельно отображаются состояния остальных МодПр.

Горизонтальной осью на ПСС-диаграмме является ось реального времени. Все события расположены на ПСС-диаграмме таким образом, что если провести на ней вертикальную линию, то все события, расположенные левее этой линии произошли в реальном времени не раньше событий, расположенных правее этой линии.

Временные диаграммы

Если из ПСС-программы выбрать только те события, которые содержат в себе информацию об изменениях значений сигналов в узлах моделируемой схемы, можно построить временную диаграмму сигналов. Пример временной диаграммы сигналов приведен на рис. 5.

Временные диаграммы сигналов предназначены для визуального изображения последовательности сигналов в узлах и на выходах моделируемой схемы. Временная диаграмма не зависит от способа моделирования, не будет изменяться при моделировании на нескольких или при моделировании на одном процессоре.

Временные диаграммы используются для визуального представления результатов моделирования и помогают оценить корректность результатов.

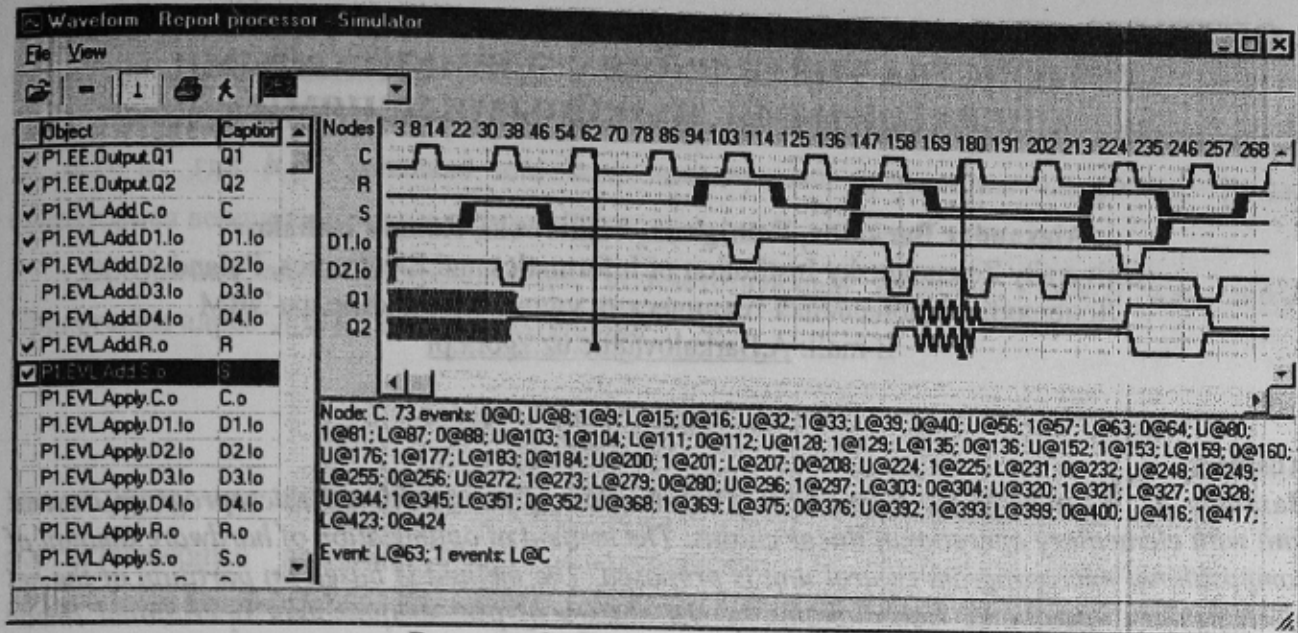


Рисунок 6 – Временные диаграммы

Выводы

Предложенную архитектуру программной системы можно использовать для исследования различных протоколов синхронизации при распределенном логическом моделировании. Детальное исследование ПСС-диаграмм позволяет определить наиболее узкие места в реализации протокола синхронизации и ошибки в реализации, которые не могут быть выявлены путем анализа временных диаграмм.

Перспективным направлением дальнейших исследований является исследование методов разрезания схем на части. Разрезание схем на части разными способами необходимо для того, что бы исследовать работу протоколов синхронизации логических процессов при различных параметрах разрезания.

Перечень источников

1. Ладыженский Ю.В. Исследование цифровых систем с программируемой логикой методами моделирования. В кн.: Научно-методические труды IX Международной научно-методической конференции. Таганрог, ТРТУ, 1999, С. 59.
2. Bashkov E.A., Ladyzhensky Y.V. Study by research in improving of EDA tools teaching in a technical university. 2nd Global Congress of Engineering Education. Wismar, Germany. Congress Proceedings, UICEE 2000, p. 462-464.
3. Ладыженский Ю.В., Попов Ю.В. Система распределенного логического моделирования цифровых устройств с использованием консервативного протокола синхронизации // Наукові праці Донецького національного технічного університету. Серія: інформатика, кібернетика та обчислювальна техніка, випуск 39. – Донецьк: ДонНТУ, 2002. – С. 21-29.
4. Ладыженский Ю.В., Онищенко Д.В. Система для исследования методов параллельного логического моделирования ПЛИС // Моделювання та інформаційні технології. Збірник наукових праць. – Київ: Національна академія наук України. Інститут проблем моделювання в енергетиці імені Г.Є Пухова, 2002. – Випуск 12. – С. 64-70.
5. Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. : Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 512 с.