

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ПРОТОКОЛОВ СИНХРОНИЗАЦИИ ПРИ РАСПРЕДЕЛЕННОМ ЛОГИЧЕСКОМ МОДЕЛИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ

Ладыженский Ю.В., Попов Ю.В.

Донецкий национальный технический университет,

кафедра ПМиИ

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

Abstract. Ladyzhenskyu Y.V., Popov Y.V. Object-oriented model of synchronization protocols for distributed logical simulation of digital devices. Architecture of a system for distributed synchronization protocols research is proposed. Object-oriented analysis of synchronization protocols for logic simulations is fulfilled. Objects, their methods and properties are outlined.

Введение. Ускорение логического моделирования сверхбольших интегральных схем может быть достигнуто путем распараллеливания вычислений. Проведенный сравнительный анализ различных протоколов синхронизации логических процессов при распределенном моделировании [1] показал их существенное влияние на качество и эффективность моделирования. Для получения наилучших результатов при моделировании дискретных логических схем необходимо исследовать работу различных протоколов синхронизации логических процессов при распределенном моделировании.

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 [2].

В статье предложена архитектура системы для экспериментального исследования разных протоколов синхронизации логических процессов при распределенном моделировании. Проведен объектно-ориентированный анализ архитектуры консервативного и оптимистического процессов моделирования.

Выделены объекты, существующие при выполнении большинства протоколов синхронизации логических процессов. Приведена диаграмма вариантов использования объектов системы моделирования, на основе которой построен стандартизованный интерфейс между объектами системы.

Архитектура системы для исследования протоколов синхронизации.

Для исследования работы различных протоколов синхронизации логических процессов при распределенном моделировании необходима специальная программная система. Эта система должна включать в себя набор утилит для ввода исходных данных, проведения моделирования, сбора информации о процессе моделирования и анализа полученных результатов [3,4]. Эти утилиты будут необходимы для анализа работы большинства существующих протоколов синхронизации логических процессов. Архитектура такой программной системы приведена на рис. 1.

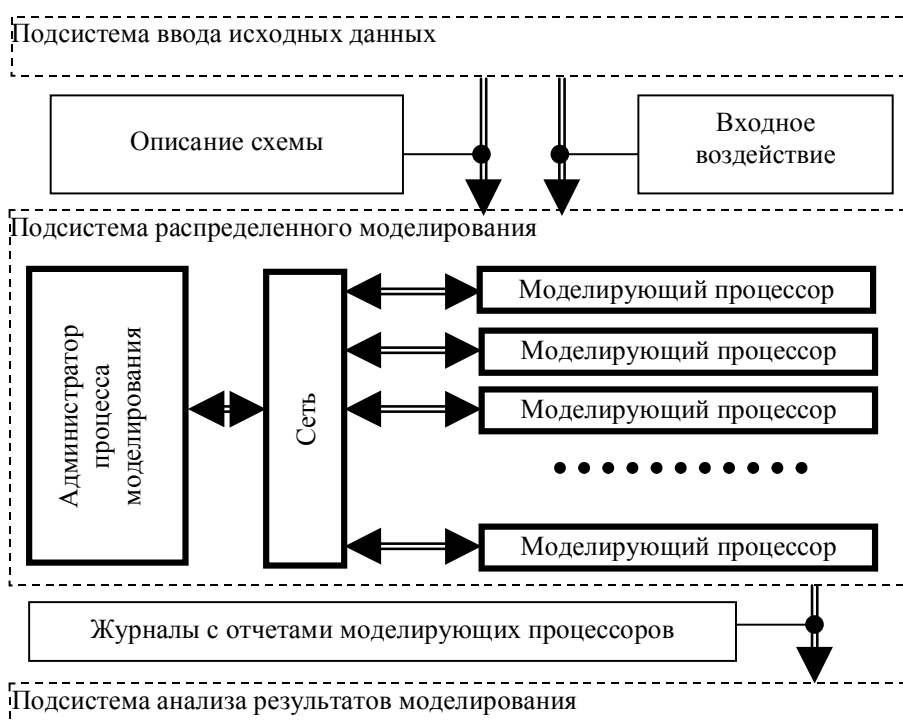


Рисунок 1 — Архитектура программной среды для исследования протоколов синхронизации логических процессов при распределенном моделировании

Моделирующий процессор (МодПр) содержит в себе:

- библиотеку сервисных классов, которые не изменяются при переходе от одного протокола синхронизации к другому. Дальнейший анализ протоколов синхронизации направлен на выявление этих классов;

- координатор процесса моделирования (КПМ) – набор классов, которые реализуют протокол синхронизации логических процессов и подсистемы, необходимые для работы данного протокола синхронизации и не нужны при работе других протоколов. Например, подсистема обнаружения и восстановления из тупиков, которая используется в консервативном протоколе и не нужна в оптимистическом протоколе.

При реализации различных протоколов синхронизации могут потребоваться одинаковые библиотеки классов. Для того, что бы определить общие для разных протоколов синхронизации объекты, их методы и свойства, нужно провести объектно-ориентированный анализ архитектур основных протоколов синхронизации логических процессов [5,6].

Объектно-ориентированный анализ архитектуры консервативного протокола синхронизации процессов.

Архитектура консервативного протокола синхронизации логических процессов приведена на рис. 2. Логические процессы LPk передают сообщения через систему коммуникаций. На каждом процессоре есть своя структурно-функциональная модель схемы (СФМС), планировщик и коммуникационный интерфейс (КИ). СФМС содержит в себе описание ориентированного графа связей между элементами Rk и значения в узлах схемы Sk. Планировщик предназначен для управления порядком обработки событий в пределах одного логического процесса. Для этого в планировщике имеются локальные часы LVT и локальный список событий LEVL. КИ предназначен для управления глобальным порядком обработки событий. Для этого планировщик задач включает в себя входные буферы IVi и выходные буферы OVi. Для каждого входного буфера в КИ содержатся каналные часы CСi, которые показывают локальное время на процессоре LPi. Поскольку события приходят строго в таком порядке, в котором были отправлены, это время равняется времени последнего пришедшего по этому каналу сообщения. На основании всех CСi КИ вычисляет значение горизонта расширения локального виртуального времени LVTH. LVTH равняется минимуму среди всех значений CСi. КИ разрешает планировщику задач выполнять только те события, время выполнения которых не превосходит значения LVTH [7,8].

Для осуществления работы консервативного протокола синхронизации логических процессов при распределенном моделировании необходимы следующие основные объекты:

- структурно-функциональная модель схемы;
- локальные часы;
- локальный список событий;
- консервативный коммуникационный интерфейс;
- система коммуникаций.

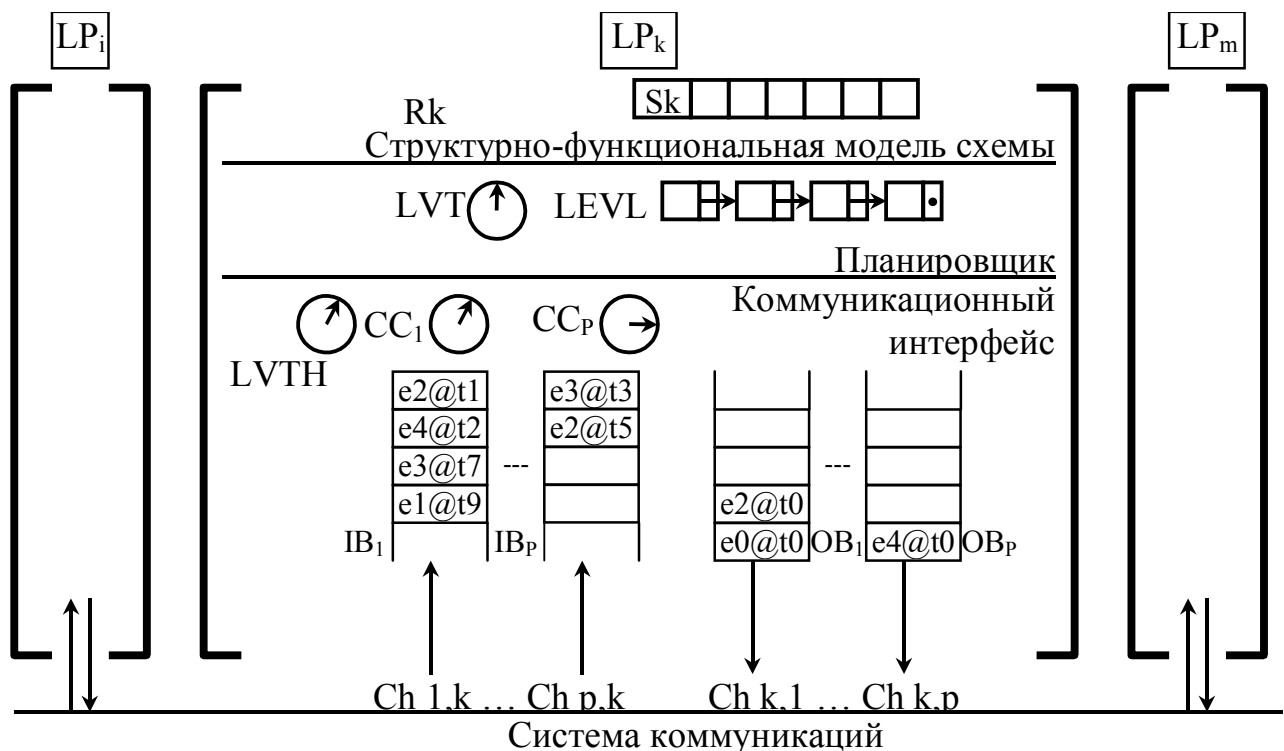


Рисунок 2 — Архитектура консервативного протокола синхронизации логических процессов

Кроме этого, возможно создание объектов для подсистем, предназначенных для осуществления корректной работы консервативного протокола синхронизации. Например, подсистема определения/восстановления системы из тупиков.

Объектно-ориентированный анализ архитектуры оптимистического протокола синхронизации процессов. Архитектура оптимистического протокола синхронизации логических процессов приведена на рис. 3. Архитектура оптимистического протокола синхронизации логических процессов аналогична

архитектуре консервативного протокола. В системе так же, как и при использовании консервативного протокола, присутствует СФМС, планировщик, КИ и система коммуникаций. Отличие от архитектуры консервативного протокола синхронизации состоит в КИ, остальные подсистемы имеют то же назначение и функционируют аналогичным образом.

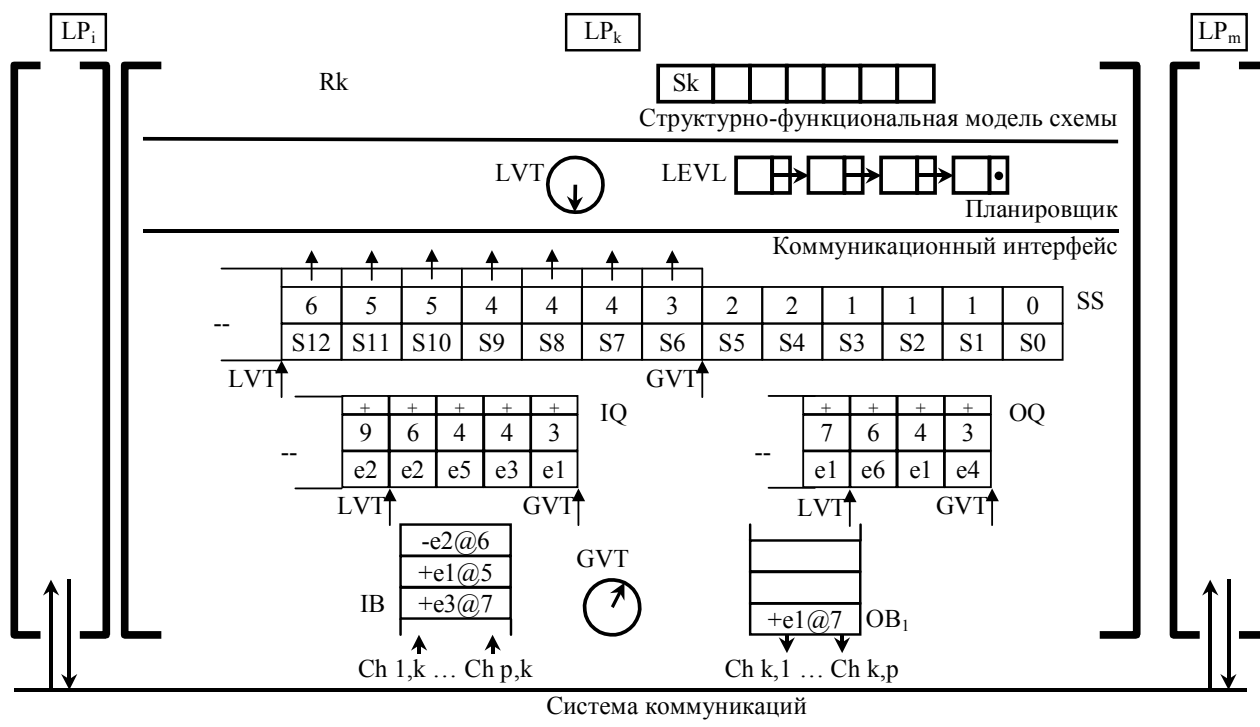


Рисунок 3 — Архитектура оптимистического протокола синхронизации логических процессов

В оптимистическом КИ содержатся часы, показывающие GVT, входной буфер IB, выходной буфер OB, входная очередь IQ, выходная очередь OQ и список всех состояний системы SS. GVT вычисляется одним из существующих алгоритмов. В отличие от консервативного протокола, при использовании оптимистического протокола сообщения не должны приходить в том порядке, в котором они были отправлены и КИ не задерживает обработку событий планировщиком. Планировщик обрабатывает все события, которые находятся в локальном списке событий или во входной очереди. К сообщениям о событиях в оптимистическом протоколе добавляется знак "+" или "-", который означает «сообщение» или «антисообщение». Если приходит сообщение, то оно записывается во входную очередь. Когда приходит антисообщение:

- соответствующее ему сообщение удаляется из входной очереди;

- производится откат состояния системы до момента наступления этого антисообщения;
- отправляются антисообщения для тех отправленных сообщений о событиях которые произошли после обработки удаленного сообщения о событии.

Не может прийти антисообщение с моментом виртуального времени меньше GVT, поэтому все сообщения о событиях из входной очереди IQ, время наступления которых меньше GVT, могут быть удалены из памяти. Для того, что бы проводить откат состояний, в коммуникационном интерфейсе содержится список всех состояний системы SS. Состояния, соответствующие виртуальному времени, меньшему, чем значение GVT, заведомо изменены не будут и могут быть удалены из памяти в журнал отчета. Для того, что бы установить, какие антисообщения нужно отправить, в коммуникационном интерфейсе содержится выходная очередь OQ. Поскольку антисообщения, соответствующие моментам времени, меньшим GVT, прийти не могут, то и антисообщения, соответствующие моментам времени, меньшим GVT, отправлены тоже не будут. Поэтому все сообщения, соответствующие моментам времени, меньшим GVT, могут быть удалены из выходной очереди [7,8].

Варианты использования объектов системы моделирования. Для того, что бы исследовать работу разных протоколов синхронизации, необходимо иметь возможность замены одного протокола синхронизации другим. Некоторые объекты используются при использовании любого протокола синхронизации, некоторые – специфичны для выбранного протокола синхронизации.

Проанализировав архитектуру консервативного и оптимистического протоколов синхронизации логических процессов, определим объекты, которые будут реализованы в сервисном модуле и в координаторе процесса моделирования. Для того, что бы легко можно было заменять один протокол синхронизации другим, необходимо разработать стандартизованный интерфейс между сервисным модулем и КПМ. Для этого рассмотрим диаграмму вариантов использования этих объектов (рис. 4).

При исследовании различных протоколов синхронизации изменяться будет не только КИ. Для обеспечения работоспособности протоколов синхронизации могут потребоваться некоторые подсистемы, специфические для данного типа протоколов синхронизации. Например, подсистема обнаружения тупиков

в консервативном протоколе и подсистема вычисления GVT в оптимистическом протоколе. КИ и специфичные для этого интерфейса подсистемы объединены в отдельный модуль, он называется «координатор процесса моделирования» (КПМ).

Общие для разных протоколов синхронизации объекты «система коммуникаций», «локальные часы», «локальный список событий» и «структурно-функциональная модель схемы» объединены в отдельный модуль. Этот модуль называется «сервисный модуль».

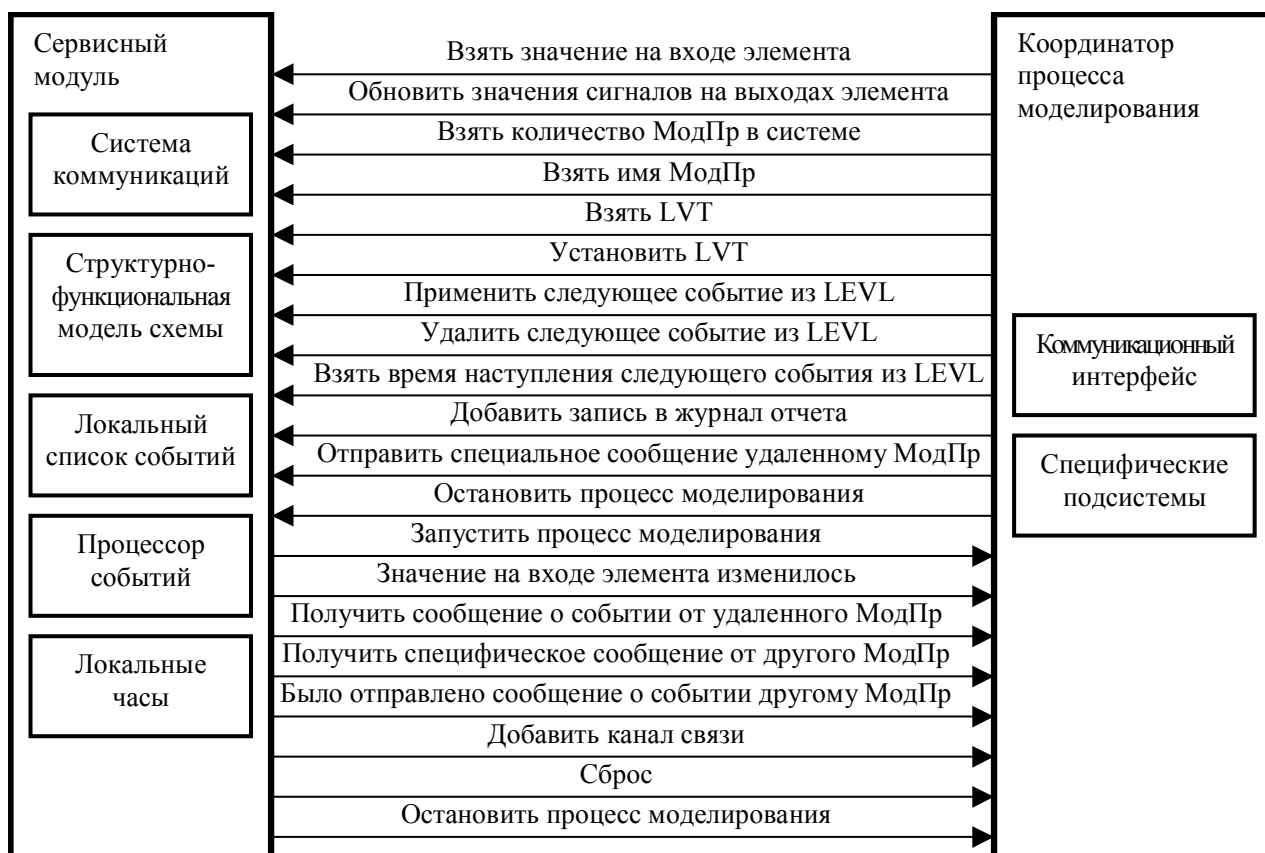


Рисунок 4 — Диаграмма вариантов использования объектов «сервисный модуль» и «координатор процесса моделирования»

Таким образом, необходимо создать библиотеку классов, которая реализует сервисный модуль. Для исследования различных протоколов синхронизации логических процессов нужно будет изменять только КПМ. КПМ вместе с сервисным модулем представляют собой моделирующий процессор (МодПр). Один МодПр соответствует одному логическому процессу.

Стрелки, направленные от КПМ к сервисному модулю (рис. 4) указывают на, методы реализуемые интерфейсе сервисного модуля. Стрелки, направленные от сервисного модуля к КПМ указывают на то, какие методы должен реализовывать в своем интерфейсе КПМ. Методы интерфейсов вызываются в следующих случаях:

- «взять значение на входе элемента» будет использоваться оптимистическим протоколом синхронизации при сохранении состояний;
- «обновить значения сигналов на выходах элемента». КИ может задержать обновление значения сигналов на выходах элемента в случае, если на входах элемента могут произойти другие события в текущий момент LVT. Такие задержки могут присутствовать как в консервативном, так и в оптимистическом протоколах синхронизации;
- «взять количество МодПр в системе» и «взять имя МодПр» используются для получения информации о МодПр в системе. Например, при использовании консервативного протокола синхронизации может существовать подсистема управления маркером, которая должна определить последовательность процессоров, по которой маркер будет циркулировать в системе. Подсистема обнаружения тупиков использует общее количество процессоров в системе для определения состояния тупика;
- «взять LVT» и «установить LVT» используется для управления локальным виртуальным временем;
- «применить следующее событие из LEVL» является командой процессору событий взять следующее событие из LEVL и установить значения в узлах схемы в соответствии с этим событием;
- «удалить следующее событие из LEVL» применяется при использовании оптимистического протокола синхронизации, когда пришло анти-сообщение, и некоторые события из LEVL должны быть удалены;
- «взять время наступления следующего события из LEVL» используется для продвижения локального виртуального времени. События из LEVL применяются только после того, как LVT установлено на момент их наступления. Если время следующего события из LEVL равно текущему значению LVT, то следующее событие применяется. Если время наступления следующего события из LEVL больше LVT, то КИ

принимает решение: либо увеличить LVT, либо задержать обработку остальных событий;

- «добавить запись в журнал отчета» используется для передачи информации из КПМ на сервер отчетов. Эта информация будет использоваться для последующего анализа поведения протокола синхронизации;
- «отправить специальное сообщение удаленному МодПр» используется специфическими подсистемами для передачи специфических сообщений. Например, при использовании консервативного протокола синхронизации может возникнуть сообщение «увеличить LVT», которого не может возникнуть при использовании оптимистического протокола синхронизации. На МодПр, получившем специальное сообщение, будет применен метод «получить специфическое сообщение от удаленного МодПр» интерфейса КПМ;
- «остановить процесс моделирования» применяется, когда КПМ определяет, что процесс моделирования завершен. В сервисном модуле этот метод отправляет специальное сообщение всем удаленным МодПр. МодПр, получившее это сообщение, применяет метод «остановить процесс моделирования» своего КПМ;
- «запустить процесс моделирования» вызывается, когда от Администратора процесса моделирования приходит специальное сообщение о том, что схема и входное воздействие уже загружены и параметры моделирования установлены;
- событие «значение на входе элемента изменилось» применяется, когда на КПМ нужно передать информацию об элементах, на выходах которых необходимо обновить значения сигналов. Когда КИ установит, что на выходах этого элемента можно обновить значения сигналов, то КПМ должен применить метод «обновить значения сигналов на выходах элемента»;
- «получить сообщение о событии от удаленного МодПр» применяется, когда от удаленного МодПр приходит сообщение о событии. КПМ должен проверить это событие и произвести определенные действия. Например, при использовании оптимистического КИ, если это пришло антисообщение, то должен произойти откат;

- «было отправлено сообщение о событии другому МодПр». Это обработчик события, который используется для обеспечения работоспособности подсистемы обнаружения тупиков. Для того, что бы определить состояние тупика, подсистема обнаружения тупиков должна иметь информацию о том, были ли отправлены или получены какие-либо сообщения о событиях после последнего посещения маркера;
- «добавить канал связи» применяется сервисным модулем для того, что бы передать на КПМ информацию о том, какие каналы связи существуют. Эта информация требуется при использовании консервативного протокола синхронизации для того, что бы вычислять значения канальных часов для всех каналов связи и LVTH;
- «сброс» применяется сервисным модулем для того, что бы КПМ перешел в начальное состояние для моделирования новой схемы.

Заключение. Для получения наилучших результатов при распределенном моделировании дискретных логических схем необходимо исследовать работу различных протоколов синхронизации логических процессов при моделировании схем с различной степенью внутреннего параллелизма, по-разному разрезая их на части, подавая на вход разные входные воздействия. Проведенный объектно-ориентированный анализ оптимистического и консервативного протоколов синхронизации логических процессов показал наличие одинаковых объектов при использовании разных протоколов синхронизации.

Для экспериментального исследования разных протоколов синхронизации логических процессов при распределенном моделировании предложен стандартизованный интерфейс между объектами сервисного модуля и объектами координатора процесса моделирования. В сервисном модуле содержатся объекты, присутствующие при использовании большинства протоколов синхронизации. В координаторе процесса моделирования содержатся объекты, специфичные для выбранного протокола синхронизации логических процессов.

Перспективным направлением дальнейших исследований является анализ потоков информации, возникающих при взаимодействии объектов с целью определения наиболее узких мест в протоколах синхронизации. Для того, что бы наиболее эффективно провести анализ различных протоколов синхронизации, необходимо разработать средства для сбора и анализа информации о ходе моделирования.

Литература

1. 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. — 282 с. — С. 21–29.
2. 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.
3. Ладыженский Ю.В. Исследование цифровых систем с программируемой логикой методами моделирования. В кн.: Наукоемкие технологии образования: Труды IX Международной научно-методической конференции. — Таганрог, ТРТУ, 1999. — С. 59.
4. 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.
5. Гради Буч. Объектно-ориентированный анализ и проектирование.
6. Боггс У., Боггс М. UML и Rational Rose / перевод с нагл. — М.: Лори, 2001. — 581 с.
7. Ferscha Alois. Parallel and Distributed Simulation of Discrete Event Systems. In Hardbound of Parallel and Distributed Computing. McGraw-Hill, 1995.
8. Chandy K. M., Misra J. Asynchronous Distributed Simulation via a Sequence of Parallel Computations. Communications of the ACM, 24(11): 198–206, November, 1981.

Сдано в редакцию: .03.2003г.

Рекомендовано к печати: д.т.н., проф. Баркалов А.А.