

ИСПОЛЬЗОВАНИЕ ДИНАМИЧЕСКОЙ СИНХРОНИЗАЦИИ В СИСТЕМЕ РАСПРЕДЕЛЕННОГО ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ

Ладыженский Ю.В., Попов Ю.В., Тесленко Г.А.
Кафедра ПМиИ, ДонНТУ

Рассмотрена структура программной системы для распределенного логического моделирования. Предложена организация подсистемы распределенного моделирования с динамическим типом синхронизации. Предложены алгоритмы переключения протокола синхронизации.

Введение. Цифровые системы логического управления на основе микросхем, программируемых пользователем, широко применяются в промышленности. Моделирование цифровых систем является важным этапом проектирования, обеспечивающим проверку корректности функционирования. Высокая размерность и сложность логических схем приводит к большим затратам времени и памяти при последовательном моделировании.

Использование алгоритмов параллельного и распределенного вычислений [1, 2] является перспективным направлением для ускорения моделирования. Создание на их основе эффективных по быстродействию параллельных и распределенных программных и аппаратных средств моделирования цифровых устройств является актуальной научно-технической проблемой.

Задачей работы является исследование динамических алгоритмов синхронизации с целью повышения быстродействия распределенного логического моделирования цифровых систем.

Динамический протокол синхронизации. Особенностью динамического протокола, используемого в системе моделирования, является возможность переключения от консервативной синхронизации к оптимистической и обратно в процессе моделирования. Логический процесс (ЛП) автоматически изменяет свой тип на консервативный в случае возникновения частых откатов, либо на оптимистический в случае частых блокировок при моделировании. Переключение в оптимистический режим синхронизации выполняется путем установки поля типа логического процесса. Переключение в консервативный режим предполагает в начале использование промежуточного режима работы, в котором

обрабатываются только безопасные и антисообщения. Так как при оптимистической синхронизации в списке состояний могут находиться события, которые потенциально могут привести к откату в других логических процессах, то перед переключением необходимо гарантировать корректное состояние логического процесса. В промежуточном режиме выполняется откат ошибочных событий с помощью антисообщений, новые состояния не сохраняются в очередь, т.к. обрабатываются только безопасные события. В результате очередь состояний очищается, что гарантирует корректное состояние логического процесса перед его переключением в консервативный режим.

С целью повышения эффективности моделирования динамической протокол синхронизации может быть реализован аппаратно [3, 4].

Структура программной системы. Программная система для распределенного логического моделирования состоит из трех подсистем [5]: подсистема подготовки схемы и входного воздействия для моделирования, подсистема моделирования и подсистема анализа результатов моделирования (рисунок 1).

Подсистема подготовки схемы и входного воздействия для моделирования предназначена для создания и редактирования схем и входного воздействия.

В подсистеме моделирования каждый моделирующий процессор выполняет моделирование заданного участка схемы с заданным воздействием. В результате моделирования создается отчет, который является входными данными для подсистемы анализа результатов моделирования. Эта подсистема позволяет построить временные диаграммы, диаграммы причинно-следственных связей и получить статистическую информацию из отчетов моделирований.

Программная система позволяет последовательно моделировать цифровые системы, либо выполнять распределенное моделирование с использованием консервативного, оптимистического, динамического алгоритмов.

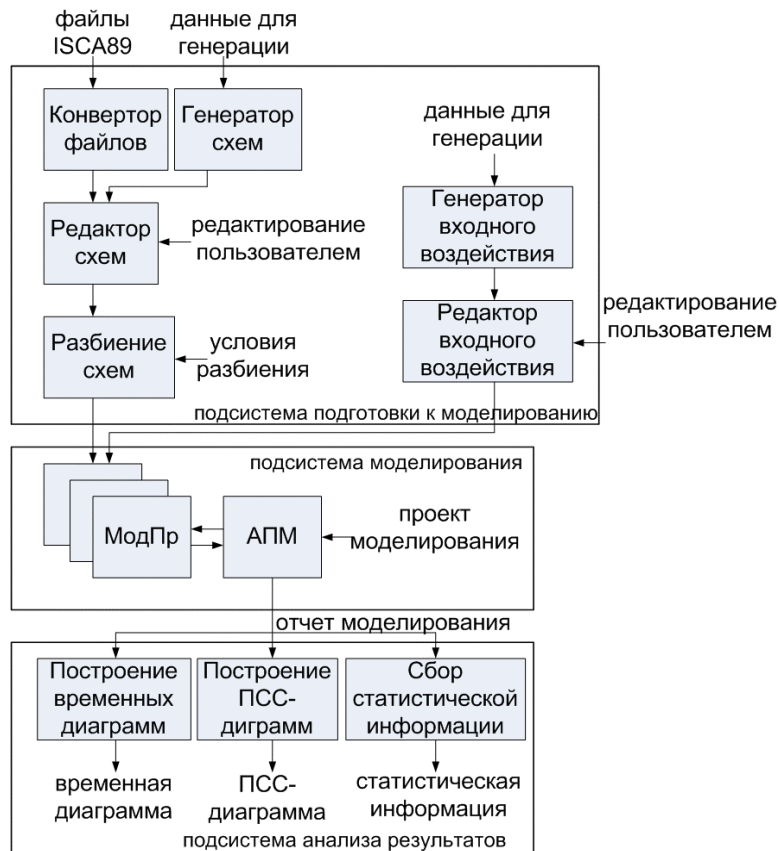


Рис. 1 – Структура программной системы для распределенного моделирования

Организация подсистемы распределенного моделирования.

Подсистема моделирования состоит из следующих программных компонент (рисунок 2):

- администратор процесса моделирования (АПМ), специальная программа для управления моделирующими процессами. Администратор позволяет осуществлять выбор схемы, входного воздействия, а также задавать необходимые опции для моделирования. При моделировании синхронизация осуществляется только средствами моделирующих процессоров, без участия АПМ. После завершения моделирования администратор собирает отчеты всех моделирующих процессоров. Для обмена данными между администратором процесса моделирования и моделирующими процессами может использоваться локальная сеть или Интернет[6].

- множество моделирующих процессоров (МодПр), каждый из которых реализует алгоритм синхронизации вычислений.

Моделирующий процессор содержит такие объекты: сервисный модуль, координаторы процесса моделирования, библиотеку функций элементов схемы.



Рис. 2 – Структура подсистемы распределенного моделирования
 Общие для разных протоколов синхронизации объекты «система коммуникаций», «локальные часы», «локальный список событий», «структурно-функциональная модель схемы» и «переключатель типа синхронизации» содержатся в сервисном модуле.

Локальный список событий содержит все события, которые возникнут в моделируемой системе. События в этом списке упорядочены в порядке увеличения виртуального времени их наступления. Локальные часы показывают текущее виртуальное время. Структурно-функциональная модель схемы содержит структуру моделируемой цифровой системы и значения сигналов во всех узлах схемы.

Переключатель типа синхронизации во время моделирования выбирает оптимальный протокол синхронизации с целью сокращения времени моделирования.

Координатор процесса моделирования (КПМ) управляет порядком работы всех объектов моделирующего процессора, содержит цикл обработки событий. В подсистеме моделирования в один момент времени может работать только один КПМ, консервативный или оптимистический. В состав каждого из КПМ входит коммуникационный интерфейс, предназначенный для управления глобальным порядком обработки событий.

Консервативный КПМ содержит коммуникационный интерфейс и подсистему выхода из тупиков. Консервативный коммуникационный интерфейс включает в себя каналные часы,

входные, выходные буферы событий для каждого канала связи с остальными МодПр.

Оптимистический КПМ содержит коммуникационный интерфейс и подсистему вычисления глобального виртуального времени GVT. В оптимистическом коммуникационном интерфейсе содержатся часы, показывающие глобальное виртуальное время, входной и выходной буфер, входная, выходная очереди событий, и список всех состояний системы.

КПМ и сервисный модуль образуют сетевой моделирующий процессор (МодПр). Один МодПр соответствует одному логическому процессу и запускается на отдельном компьютере.

Алгоритмы переключения типа синхронизации в подсистеме моделирования. Динамический тип синхронизации реализуется за счет переключения в процессе моделирования между оптимистическим и консервативным алгоритмами [7]. Для переключения типа синхронизации необходимо запустить соответствующий координатор процесса моделирования. На рисунках 3 и 4 показаны алгоритмы переключения протоколов синхронизации в подсистеме распределенного моделирования из консервативного в оптимистический и обратно.

КПМ представляет собой вычислительный поток операционной системы, запускаемый из сервисного модуля. При запуске МодПр в памяти динамически создаются все его объекты, включая консервативный и оптимистический координаторы процесса моделирования в виде двух отдельных вычислительных потоков. По умолчанию запускается КПМ, указанный в проекте моделирования, КПМ другого типа находится в «спящем» состоянии.

Переключение консервативного МодПр в оптимистический выполняется следующим образом. Инициатор переключения рассылает всем сообщения о необходимости переключения. Далее, все МодПр сохраняют состояние своего коммуникационного интерфейса (внешние события входной и выходной очередей). Затем моделирование останавливается, путем остановки вычислительного потока КПМ. Каждый МодПр рассылает всем подтверждения о готовности к переключению типа синхронизации. После принятия подтверждения от всех МодПр выполняется запуск и установка начальных параметров оптимистического КПМ. Происходит изменение состояния коммуникационного интерфейса (заполнение входной, выходной очередей внешними событиями). Далее



Рис. 3 – Алгоритм переключения консервативного протокола синхронизации в оптимистический



Рис. 4 – Алгоритм переключения оптимистического протокола синхронизации в консервативный

моделирование продолжается в соответствии с оптимистическим типом синхронизации.

Если МодПр решает стать консервативным, он также рассылает всем сообщения о необходимости переключения. После получения такого сообщения МодПр переходят в промежуточный режим работы, в котором обрабатываются только безопасные и антисообщения. Далее алгоритм соответствует описанному выше.

Заключение. Рассмотрена структура существующей программной системы для распределенного логического моделирования. Предложена организация подсистемы распределенного моделирования с динамическим типом синхронизации. Описаны алгоритмы переключения типа синхронизации.

Использование алгоритма динамической синхронизации позволяет повысить эффективность распределенного логического моделирования по сравнению с классическими алгоритмами.

В дальнейшем планируется провести исследование критерия переключения типа синхронизации.

Литература

1. C.J.R. Shi, D.Lungeanu. Distributed simulation of VLSI circuits via lookahead-free self-adaptive optimistic and conservative synchronization. In Proc. ICAAD, pages 500-504, Nov 1999.

2. D.Lungeanu and C.-J.R. Shi. Parallel and distributed vhdl simulation. In Proc. DATE, pages 658-662, March 2000.

3. Ладыженский Ю.В., Тесленко Г.А. Аппаратный метод повышения эффективности алгоритмов распределенного логического моделирования цифровых систем. // Наукові праці Донецького національного технічного університету. Серія: «Обчислювальна техніка та автоматизація». Випуск 106 – Донецьк: ДонНТУ, 2006. – 220с. – С.77-81.

4. Ладыженский Ю.В., Тесленко Г.А. Аппаратные методы повышения эффективности распределенного логического моделирования. // Интернет-Освіта-Наука-2006, п'ята міжнародна конференція ІОН-2006, 10-14 жовтня, 2006. Збірник матеріалів конференції. Том 2. – Вінниця: УНІВЕРСУМ-Вінниця, 2006. – 420с. –С.746-749.

5. Ladyzhensky Y.V., Popoff Y.V. Software system for event-driven logic simulation // IEEE EWDWT, Odessa, September 15-19, 2005, p.119-122

6. Ladyzhensky Y.V., Popoff Y.V. Architecture of internet access to distributed logic simulation system // IEEE EWDWT, Sochi, September 15-19, 2006, p.339-343

7. Ладыженский Ю.В., Попов Ю.В., Тесленко Г.А. Программная система для распределенного логического моделирования с динамическим протоколом синхронизации // Радиоэлектронные и компьютерные системы. – 2007. - №8(27). – С.25-29.

Получено 01.06.07