

## АППАРАТНЫЙ МЕТОД ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ РАСПРЕДЕЛЕННОГО ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ

Ладыженский Ю.В., Тесленко Г.А.

Донецкий национальный технический университет, г. Донецк  
кафедра прикладной математики и информатики  
E-mail: ly@cs.dgtu.donetsk.ua

### *Abstract*

*Ladyzhensky Y.V., Teslenko G.A. Hardware method to increase efficiency of algorithms for distributed logic simulation of digital systems. A method of hardware increase of logic simulation efficiency of the digital systems is proposed. A structural implementation of combined protocol with the dynamic synchronization of calculations and structure of specialized processor is presented.*

### **Введение**

Сложность и размеры проектов цифровых систем постоянно увеличиваются. Одновременно совершенствуются программные системы верификации проектов, использующие логическое событийное моделирование [1]. Хотя производительность систем верификации постоянно повышается, ее увеличение отстает от роста размеров и сложности разрабатываемых проектов СБИС. Поэтому системы моделирования должны быть реализованы на аппаратном уровне. Аппаратное ускорение способно значительно увеличить скорость моделирования. Применение такого подхода для логического событийного моделирования ранее в достаточной мере не рассматривалось. Целью данной работы является исследование аппаратного метода повышения эффективности распределенного логического моделирования для проектов цифровых систем с большим уровнем интеграции.

### **Распределенное логическое моделирование**

Имитационное моделирование является важным инструментом при верификации проектов цифровых систем. Оно используется для функционального и временного моделирования проекта и проводится с целью выявления ошибок, возникающих на стадиях проектирования.

Распределенная система моделирования состоит из процессов, которые взаимодействуют друг с другом путем передачи сообщений [2]. Каждый процесс моделирует часть физической системы и представляет собой логический процесс (ЛП). В течение времени моделирования ЛП генерируют события, посылают их другим логическим процессам и получают события от других процессов. Распределенное моделирование считается корректным, если каждый логический процесс обрабатывает события в порядке возрастания значений их временных меток. Существует два типа алгоритмов синхронизации вычислений, консервативные и оптимистические, которые широко используются при распределенном моделировании.

Распределенная система может состоять из рабочих станций, объединенных в локальную сеть (рис.1). Исходная модель цифровой системы на языке аппаратного описания преобразуется компилятором в список соединений. Полученный список затем используется блоком разбиения для разрезания исходного графа схемы на подграфы. В результате разрезания полученные участки схемы распределяются для моделирования между специализированными рабочими станциями (СРС). Каждая рабочая станция содержит процессор распределенного моделирования (ПРМ), реализованный в виде стандартной платы расширения.

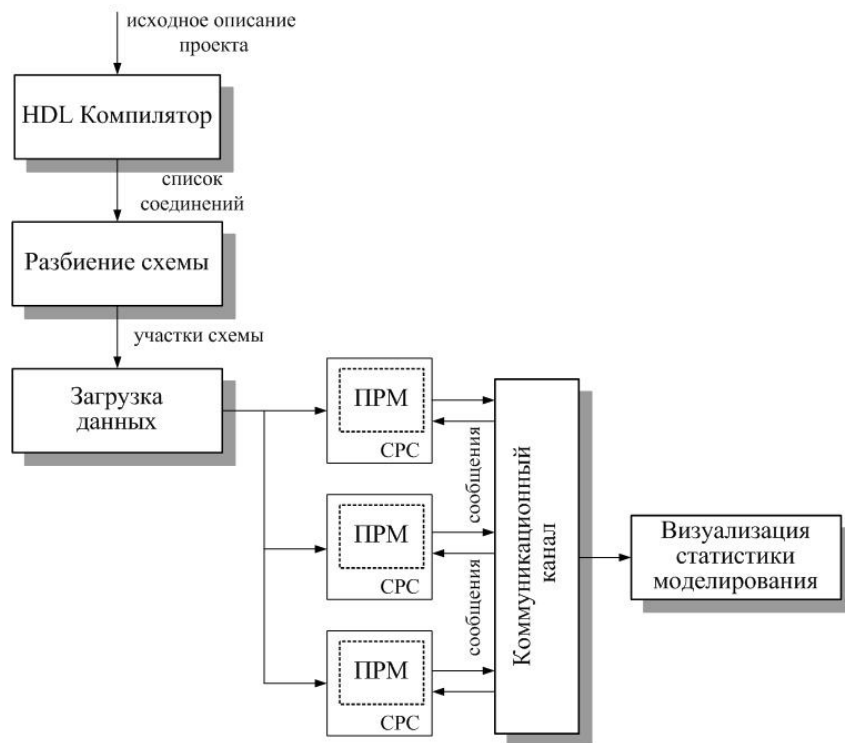


Рисунок 1 — Структура распределенной системы моделирования

**Аппаратный метод реализации алгоритмов распределенного моделирования**

Программный метод интерпретации алгоритмов не является единственным способом решения задач. Можно применять структурную или аппаратную реализацию алгоритма [3]. Такая реализация заключается в создании из набора функциональных блоков и связей между ними структурной модели, обеспечивающей эквивалентное отображение схемы алгоритма и процесса обработки данных. Алгоритм реализуется за счет продвижения данных и их преобразования по путям обработки (асинхронно или при тактировании) от входов к выходам схемы, на которых получается результат. Команды для выполнения отдельных операций могут отсутствовать. Это обеспечивает распределение вычислительного процесса не только во времени, но и в пространстве. Естественным образом достигается высокая степень параллелизма. Необходимый набор функциональных блоков можно реализовать на микросхемах программируемой логики (ПЛИС), которые в настоящее время широко применяются для построения цифровых систем.

**Структурная реализация процессора распределенного моделирования**

Рассмотрим структурную реализацию распределенного моделирования с применением комбинированного алгоритма синхронизации вычислений. Главным его достоинством является динамическая синхронизация, т.е. возможность переключения от оптимистического поведения к консервативному и обратно [4,5]. Оптимистический ЛП может стать консервативным в случае слишком частых откатов, и наоборот, консервативный становится оптимистическим в случае частых блокировок выполнения. Структурная схема процессора распределенного моделирования приведена на рисунке 2. Ниже приводится классификация событий и сообщений, используемых при работе процессора моделирования:

положительное событие — содержит информацию о значении сигнала в узле моделируемой схемы;

отрицательное событие — используется для отмены результатов ошибочного моделирования в случае отката;

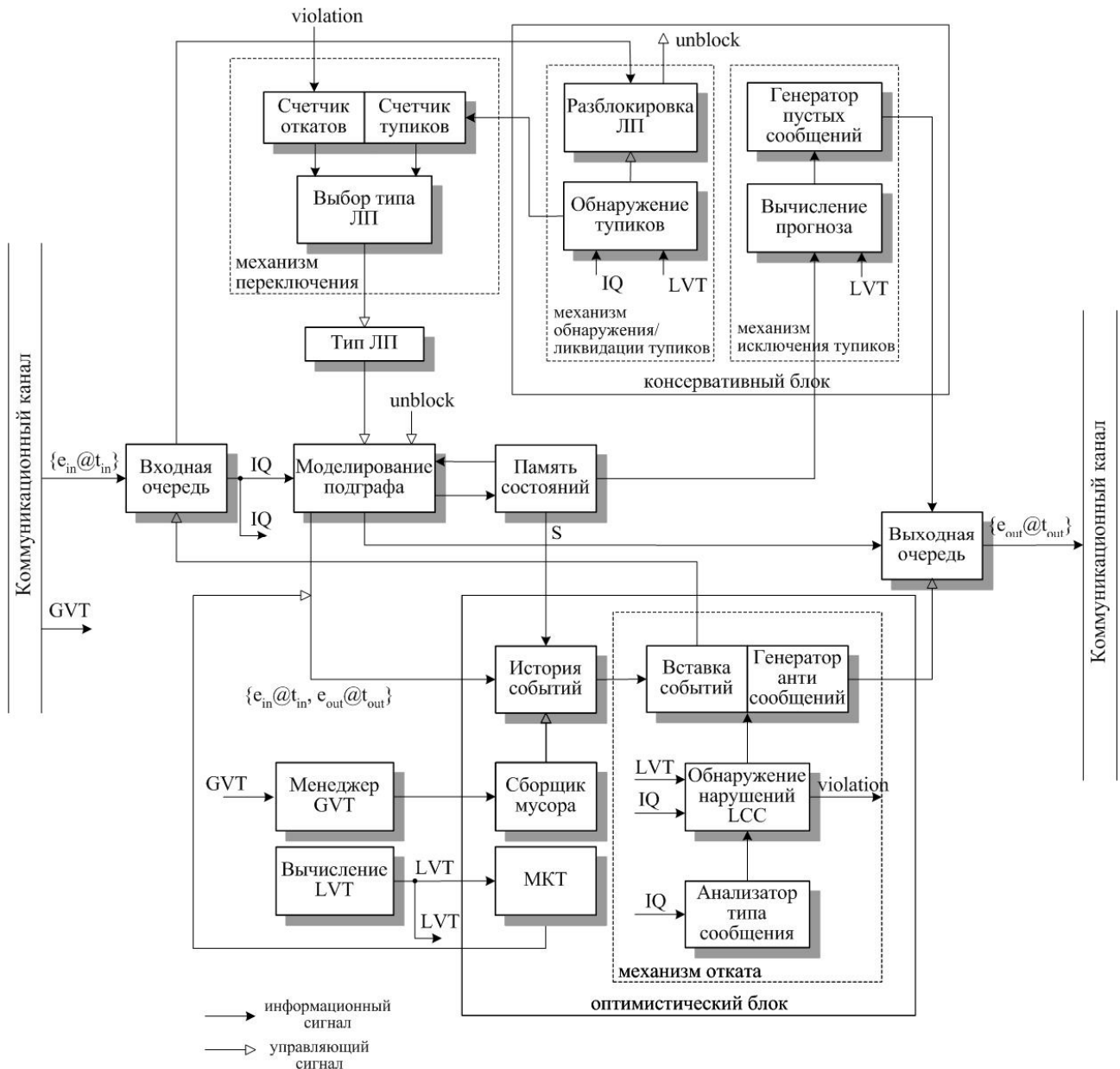


Рисунок 2 — Структурная схема процессора распределенного моделирования

безопасное событие — положительное событие, не нарушающее очередность обработки событий в локальном времени ЛП;

пустое сообщение — используется для продвижения локального виртуального времени ЛП;

антисообщение — содержит информацию об отрицательном событии.

Каждому событию соответствует время его возникновения:

$\{e_{in}@t_{in}\}$  — входное событие  $e_{in}$ , сформированное во время  $t_{in}$ , принимаемое для обработки ЛП;

$\{e_{out}@t_{out}\}$  — выходное событие  $e_{out}$ , сформированное во время  $t_{out}$ , передаваемое для обработки остальным ЛП;

GVT (Global virtual time) — глобальное время моделирования;

LVT (Local virtual time) — локальное время моделирования.

В рассматриваемой структуре блок входной очереди служит для приема событий  $\{e_{in}@t_{in}\}$ , принятых из канала коммуникации.

Блок моделирование подграфа схемы цифрового устройства принимает сообщения из входной очереди IQ (input queue), обрабатывает их, формирует выходные сообщения  $\{e_{out}@t_{out}\}$  и помещает их в выходную очередь, откуда они передаются в коммуникационный канал.

Способ обработки входящих событий зависит от блока “Тип ЛП”, который определяет, является ли логический процесс оптимистическим или консервативным.

Блок памяти состояний представляет собой очередь, которая хранит информацию  $\{S\}$  о состояниях выполняемого ЛП.

Консервативный блок структуры состоит из двух блоков. Блок механизма исключения тупиков предназначен для предупреждения возникновения тупиковых ситуаций в случае консервативного выполнения ЛП. При работе механизма используется протокол нулевых сообщений. В начале блок вычисления прогноза, на основании текущего состояния ЛП и локального времени процесса предсказывает минимальное время возникновения будущего события. Затем эта информация используется генератором пустых сообщений для передачи ее остальным ЛП. Блок механизма обнаружения/ликвидации тупиков используется в том случае, если уже возникла тупиковая ситуация при выполнении ЛП. В результате его работы происходит обнаружение тупиковой ситуации и разблокирование логического процесса по сигналу unblock.

Оптимистический блок реализует одноименный алгоритм синхронизации. Блок механизма отката используется для восстановления состояния ЛП в случае появления события с временной меткой меньшей, чем значение LVT. В его состав входит четыре блока. Блок анализатора типа сообщения производит проверку информации о событии, которое содержится в сообщении из входной очереди. Если событие является положительным, то происходит проверка времени его возникновения. Сигнал violation становится активным в случае нарушения хронологической последовательности обработки событий (local causality constraint, LCC). В результате отката восстанавливается содержимое памяти состояний, а неправильно промоделированные события заново вставляются во входную очередь. При этом используется информация, сохраняемая в истории событий. Генератор антисообщений вставляет в выходную очередь сообщения для отмены результатов неправильного моделирования. Блок сборщика мусора служит для удаления устаревшей информации о событиях. Блок механизма контрольных точек МКТ реализует алгоритм сохранения информации в блоке истории событий.

Блок механизма переключения реализует переключение ЛП из консервативного режима выполнения в оптимистический и обратно. В его состав входят счетчики откатов и тупиков, информация из которых используется блоком выбора типа ЛП. Переключение в оптимистический режим происходит путем установки требуемого значения в блоке “Тип ЛП”. Переключение в консервативный режим происходит несколько сложнее, поскольку возможен откат. Когда оптимистический процесс решает стать консервативным, он переключается в режим, при котором он обрабатывает только безопасные события и антисообщения. Когда история событий будет содержать только положительные события, ЛП безопасно переключается в консервативный режим.

Менеджер GVT определяет глобальное виртуальное время моделирования, которое используется всеми ЛП. Для этого используется информация, полученная из коммуникационного канала. Блок вычисления LVT обеспечивает расчет локального времени моделирования ЛП. На рисунке 3 показана структура сообщения, используемого при взаимодействии ЛП.

Адресат	Временна метка	Тип события	Информация события
---------	----------------	-------------	--------------------

Рисунок 3 — Структура передаваемого сообщения в системе

Поле Адресат содержит номер ЛП, которому адресовано сообщение. Поле Временная метка содержит информацию о времени возникновения данного события. В поле Тип события задается тип передаваемого события: положительное, отрицательное, нулевое.

Значение поля Информация события зависит от типа события. Если передается отрицательное событие, то поле содержит номер события, которое нужно отменить. В случае с положительным событием, передается информация для блока моделирования подграфа. Поле не содержит никакой информации, в случае передачи нулевого сообщения.

Использование алгоритма с динамической синхронизацией вычислений обеспечивает лучшую производительность по сравнению с консервативным, оптимистическим или комбинированным алгоритмами.

Аппаратная реализация комбинированного алгоритма позволяет существенно повысить эффективность логического моделирования цифровых систем. Так как каждому оператору алгоритма в структурной схеме ставится в соответствие свой функциональный блок и в общем случае они работают параллельно, то выигрыш по быстродействию получается пропорциональным количеству функциональных блоков.

Реализация специализированного процессора на ПЛИС позволяет создать высокоэффективные средства для распределенного моделирования цифровых устройств.

На основе описанного специализированного процессора возможно построение многопроцессорной системы моделирования. Поставив в соответствие каждому ЛП специализированный процессор и объединив их в одном комплексе, можно построить специализированную вычислительную систему для логического моделирования цифровых систем.

Использование системы процессоров распределенного моделирования с персональной ЭВМ обеспечивает удобный интерфейс пользователя.

### **Заключение**

В работе предложен метод аппаратного повышения эффективности распределенного логического моделирования цифровых систем. В представленной структурной реализации используется комбинированный протокол с динамической синхронизацией логических процессов. Выбор такого алгоритма синхронизации и его аппаратная реализация могут повысить эффективность распределенного моделирования и значительно ускорить процесс верификации проектов интегральных схем сверхбольшого уровня интеграции. Таким образом, предложенный метод является перспективным для дальнейшей разработки и исследований.

### **Литература**

1. Ладыженский Ю.В., Попов Ю.В. Система распределенного логического моделирования цифровых устройств с использованием консервативного протокола синхронизации // Наукові праці Донецького національного технічного університета. Серія: інформатика, кібернетика та обчислювальна техніка, випуск 39: — Донецьк:ДонНТУ,2002. — 282 с. — С. 21–29.
2. Ferscha. Parallel and distributed simulation of discrete event systems. Parallel and Distributed Computing Handbook, McGraw–Hill, 1995.
3. Динамическая теория информации. Основы и приложения/ В.П. Боюн. — Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2001. — 326 с.
4. 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.
5. D.Lungeanu and C.–J.R. Shi. Parallel and distributed vhdl simulation. In Proc. DATE, pages 658–662, March 2000.