

неравномерное распределение токов между модулями ПН в энергопреобразующем устройстве. В связи с тем что исследуемый тип квазирезонансного преобразователя имеет ограничение по диапазону рабочих токов нагрузки, для его расширения при сохранении высоких показателей по удельной мощности рекомендуется использовать выходной фильтр с нелинейной индуктивностью [9].

Библиографический список

1. Базилевский, А. Б. Теория и расчет автономных систем электроснабжения / А. Б. Базилевский, М. В. Лукьяненко ; Завод-втуз – филиал Краснояр. политехн. ин-та. Красноярск, 1988.
2. Системы электропитания космических аппаратов / Б. П. Соустин, В. И. Иванчура, А. И. Чернышев, Ш. Н. Ислаев. Новосибирск : Всерос. об-ние «Наука», 1994.
3. Волович, Г. Резонансные преобразователи напряжения / Г. Волович // Схемотехника. 2003. № 8. С. 10–12.
4. Горяшин, Н. Н. Исследование энергетических характеристик квазирезонансных преобразователей с целью их использования в системах электроснабжения космических аппаратов / Н. Н. Горяшин, А. Б. Базилевский // Вестник Сиб. гос. аэрокосмич. ун-та : сб. науч. тр. / Сиб. гос. аэрокосмич. ун-т. Вып. 5. Красноярск, 2004. С. 63–69.
5. Горяшин, Н. Н. Практика создания квазирезонансных преобразователей напряжения / Н. Н. Горяшин // Электронные и электромеханические устройства : сб. науч. тр. Новосибирск : Изд-во Сиб. отд-ния Рос. акад. наук, 2007. С. 121–130.
6. Лукин, А. В. Квазирезонансные преобразователи постоянного напряжения / А. В. Лукин // Электропитание. 1993. Вып. 2. С. 24–37.
7. Batarseh, I. Unified Steady-State Analysis of Soft-Switching DC-DC Converters / I. Batarseh, J. Abu-Qahouq / IEEE Trans. Power Electronsep. 2002. Vol. 17, № 5. P. 684–691.
8. Characterization and Comparison of Noise Generation for Quasi-Resonant and Pulsedwidth-Modulated Converters / L. Hsiu, M. Goldman, R. Carlsten, et al. // IEEE Trans. Power Electronics. 1994. Vol. 9, № 4. P. 425–432.
9. Goryashin, N. The Application of Output Filter with Variable Inductance in Zero-Current-Switch Quasi-Resonant Converter / N. Goryashin // J. of Electrical Engineering. 2008. Vol. 8. P. 78–83.
10. Erickson, R. W. Fundamentals of Power Electronics / R. W. Erickson. First Ed. N. Y. : Chapman and Hall, 1997.
11. Разевиг, В. Д. Схемотехническое моделирование с помощью MicroCap 7.0 / В. Д. Разевиг. М. : Горячая линия–Телеком, 2003.

N. N. Goryashin, M. V. Lukyanenko, A. A. Solomatova, A. Y. Khoroshko

MODELING OF PARALLEL OPERATION OF ZERO-CURRENT SWITCHING QUASI-RESONANT CONVERTERS

P-spice simulation results of two zero-current switching quasi-resonant converters operation with parallel connection in closed loop controlled voltage regulator under frequency control are presented.

Keywords: quasi-resonant voltage transformer; dongle, frequency-impulse modulation, specific power; dynamic current alignment.

© Горяшин Н. Н., Лукьяненко М. В., Соломатова А. А., Хорошко А. Ю., 2009

УДК 519.688

К. В. Богданов, А. Н. Ловчиков

ПОСТРОЕНИЕ EDA-СИСТЕМЫ НА ОСНОВЕ СИНХРОНИЗИРОВАННЫХ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ

Предложен новый подход к созданию EDA-систем, предназначенных для моделирования схем, содержащих элементы, функционирование которых связано с резким изменением параметров и при математическом описании приводит к существенному возрастанию производных фазовых переменных, что при традиционном моделировании ведет к срыву вычислительного процесса.

Ключевые слова: моделирование, параллельные вычисления, EDA-системы.

Моделирование и анализ работы электронного оборудования – весьма сложная задача, для решения которой активно используется специализированное программное обеспечение – EDA-системы. Их развитие идет по большей части экстенсивным путем, начиная от сис-

тем, созданных в 70-х гг. прошлого столетия. Как правило, улучшаются пользовательские интерфейсы, расширяются базы данных электронных компонентов и т. п., в то время как основные вычислительные алгоритмы остаются прежними. Чаще всего все сводится к решению получив-

шейся системы дифференциальных уравнений, являющейся математической моделью моделируемого устройства. Численные методы решения при этом позволяют получать весьма точные результаты, но при моделировании систем с большим количеством электронных компонентов возникают проблемы. Это приводит к усложнению математической модели и при моделировании систем с существенными нелинейностями значительно огрубляются результаты моделирования, что часто приводит к срыву вычислительного процесса.

В связи с этим предлагается диаметрально противоположный подход: каждый блок либо компонент устройства должен быть смоделирован отдельно. Полученные модели будут выполняться внутри отдельных вычислительных процессов. При этом, если мы рассматриваем моделирование в диапазоне времени, то каждый из таких вычислительных процессов должен получать на один или несколько своих логических входов некоторые параметры (например, мгновенное значение напряжения относительно принятой «земли»). Также этот процесс должен выдавать результаты обработки входных параметров в качестве выходных данных.

Основные проблемы, связанные с такой схемой работы, заключаются в том, что необходимо обеспечить высокую степень взаимной изолированности вычислительных процессов, сохраняя возможность синхронизированного обмена данными. Можно провести аналогию с современными вычислительными сетями, где каждая вычислительная машина максимально «самостоятельна», но имеет возможность обмениваться данными, разбитыми на пакеты, с любой другой машиной в сети в произвольный момент времени. Однако существенным отличием здесь будет являться то, что каждый процесс должен выдавать и принимать порцию информации строго по синхронизирующему сигналу. В том случае, если процесс не успевает по каким-либо причинам это сделать, возможны два подхода: ожидание и уничтожение. В первом случае ни одна порция данных от других процессов не будет принята, и не будет передана, пока от всех процессов модели не будет получен ответ. Во втором случае процессы, данные от которых не получены по истечении

установленного времени, будут уничтожены либо перезапущены, а недостающие значения данных будут заменены на нулевые. Возможна и гибридная стратегия, когда процесс уничтожается после некоторого ожидания.

Реализация асинхронного обмена сообщениями, как это сделано, например, в вычислительных сетях на основе технологии Ethernet, может повлечь за собой серьезную проблему: результаты моделирования будут зависеть от производительности системы, и без предварительного профилирования реализовать модель не получится.

Как при асинхронном, так и при синхронном обмене необходим отдельный процесс-маршрутизатор. В его функции входит сбор данных от остальных процессов, уничтожение процессов, не выславших данные в течение отведенного времени, рассылка данных по процессам в соответствии с таблицей взаимосвязей (рис. 1).

Техническая реализация в этом случае требует наличия среды с определенными возможностями. Во-первых, в ней должно одновременно (параллельно) исполняться большое количество легковесных вычислительных процессов, которые будут реализовывать атомарные блоки системы. Во-вторых, она должна поддерживать обмен сообщениями между этими легковесными процессами.

Наиболее подходящей для реализации средой представляется Erlang компании Ericsson – функциональный язык программирования, позволяющий разрабатывать программное обеспечение для разного рода распределенных систем. Язык включает в себя средства порождения параллельных процессов и их коммуникации с помощью отправки асинхронных сообщений. Программа транслируется в байт-код, исполняемый виртуальной машиной, что обеспечивает переносимость. Функциональная парадигма позволяет Erlang избежать таких традиционных для императивных языков проблем распределенных приложений, как необходимость синхронизации, опасность возникновения тупиков и гонок.

Запущенный экземпляр эмулятора Erlang называется узлом. Узел имеет имя и «знает» о существовании других узлов на данной машине или в сети. Создание и взаимодействие процессов разных узлов не отличается от взаимодействия процессов внутри узла. Для создания про-

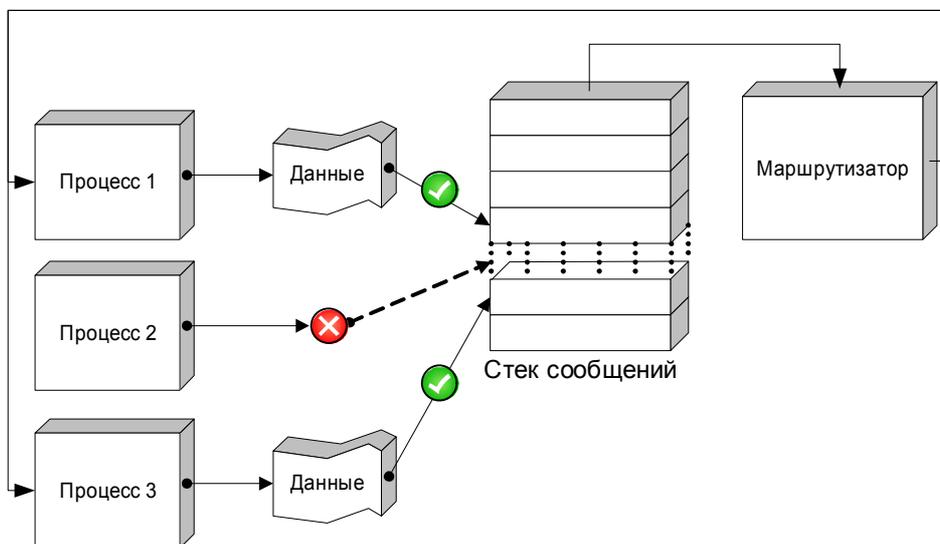


Рис. 1. Функционирование модели на основе независимых процессов

цесса на другом узле процессу достаточно знать его имя и без особых на то оснований он может не интересоваться физическим расположением взаимодействующего с ним процесса. Этим обуславливается высокая масштабируемость и способность почти линейного повышения производительности с ростом мощности системы (кластера).

Процессы, моделирующие каждый отдельный элемент схемы, смогут обмениваться сообщениями (кортежами) вида

```
{<имя корневого процесса>, <номер элемента>, <номер вывода>, <тип сигнала>, <величина сигнала 1> ..., <тип сигнала n>, <величина сигнала n>}
```

Простейший процесс, моделирующий элемент схемы, на Erlang описывается следующим образом (*Server_Node* – заранее определенное имя сервера, *modeling* – имя исполняемой программы):

```
element(Server_Node) ->
    receive
    stop ->
        exit(normal);
    {pin_number, signal1_type, signal1} ->
        % обработка входных значений %
        {modeling, Server_Node}!{self(), element_number, pin_number, signal1_type, signal1 }
    end.
```

Шаблон маршрутизатора выглядит следующим образом (*Element_List* – список всех элементов):

```
server(Element_List) ->
    receive
    {element_number, pin_number, signal1_type, signal1} ->
        % описание таблицы маршрутизации %
        element_number ! {pin_number, signal1_type, signal1 }
    end.
```

Разумеется, количество рассылаемых сообщений будет зависеть исключительно от топологии моделируемой системы, в примере показан простой случай без множественных соединений типа «один выход – много входов».

Очевидно, что для функционирования системы необходимо иметь несколько различных типов элементов: обычный процесс, коммутатор, источник, выход.

Обычный процесс (передаточная функция, пример реализации приведен выше). Обеспечивает преобразование входного потока данных в выходной. В каждый момент синхронизации обязан принять и передать одну порцию (кортеж) данных.

Коммутатор. Обеспечивает перераспределение сигналов по нескольким выходным каналам в зависимости

ти от соотношения количества входов и выходов. По сути является аналогом узла в электротехнике, но во избежание путаницы в терминологии будем называть его коммутатором. Простейший пример симметричного (распределяющего входные сигналы равномерно) коммутатора 2 на 3 будет выглядеть следующим образом:

```
commutator(Server_Node) ->
    receive
    stop ->
        exit(normal);
    {pin1, signal1_type, signal1},{pin2, signal2_type, signal2} ->
        signal1_type -> signal1_type, signal2_type, signal1_type
        (signal1+signal2)/3-> signal1, signal2, signal3
        {modeling, Server_Node}!{self(), output1, 1, signal1_type, signal1},{self(), output2, 2, signal2_type, signal2},{self(), output3, 3, signal3_type, signal3 }
    end.
```

Источник. Обеспечивает выдачу сигналов. Входов не имеет. Простейший пример источника с одним выводом:

```
commutator(Server_Node) ->
    receive
    stop ->
        exit(normal);
    {} ->
        {modeling, Server_Node}!{self(), output1, 1, type1, 100}
    end.
```

Выход. Псевдоблок, необходимый для получения текущих значений параметров для анализа. Является адаптером между моделью и пользовательским интерфейсом. Имеет один вход, выходов нет.

Одной из важнейших процедур при создании подобной модели является построение таблицы маршрутизации по пользовательской модели (к примеру, по принципиальной электрической схеме). На основе этой информации также должны создаваться процессы-коммутаторы.

После запуска все элементы начнут передавать сообщения маршрутизирующему процессу (серверу), который будет их распределять, опираясь на заданную топологию моделируемой системы.

В целом, программный продукт, построенный на основе данной схемы, будет состоять из следующих компонентов (рис. 2):

1. Ядро системы. Виртуальная машина Erlang. Основные функции – моделирование.

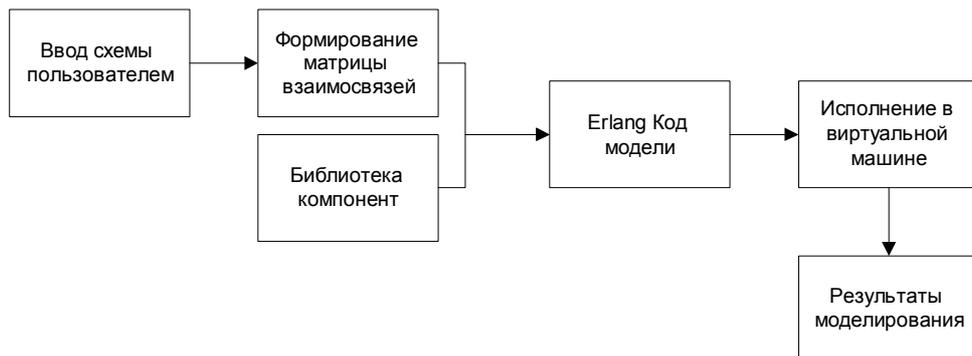


Рис. 2. Процесс моделирования

2. Библиотека компонентов. Хранилище процедур на декларативном языке Erlang.

3. Построение модели. Приложение на императивном языке высокого уровня (например, C#), формирующее в автоматическом режиме матрицу взаимосвязей для маршрутизатора и исходный Erlang-код всех процессов модели.

4. Графический интерфейс. Приложение на императивном языке высокого уровня, обеспечивающее ввод и вывод информации для пользователя.

Реализация данной системы позволит улучшить точность и скорость моделирования больших систем с существенными нелинейностями, а также позволит проводить моделирование на длительных отрезках времени без риска потерь данных.

K. V. Bogdanov, A. N. Lovchikov

CREATING EDA SYSTEMS BASED ON SYNCHRONOUS PARALLEL COMPUTING

In this article a new approach in modeling EDA systems is offered, containing elements the functioning of which is connected to sharp parameter changes; during a mathematical description it leads to an essential increase of the derivative phase variables. In traditional modeling this results in the failure of the computing process.

Keywords: modeling, parallel calculations, EDA systems.

© Богданов К. В., Ловчиков А. Н., 2009

УДК 004.4:528.9

О. Э. Якубайлик, А. А. Кадочников, В. Г. Попов, А. В. Токарев

МОДЕЛЬ ГЕОИНФОРМАЦИОННОЙ АНАЛИТИЧЕСКОЙ ИНТЕРНЕТ-СИСТЕМЫ ДЛЯ АНАЛИЗА СОСТОЯНИЯ И ПРЕЗЕНТАЦИИ РЕГИОНА*

Рассматривается модель геоинформационной интернет-системы, предназначенной для анализа состояния территории по набору показателей. Система определена как специализированный веб-портал, содержащий коллекцию типизированных компонентов-фреймов для динамического формирования веб-страниц, с элементами презентационной графики, интерактивными таблицами и картодиаграммами, с возможностью построения оценок состояния территорий в полуавтоматическом режиме.

Ключевые слова: ГИС, система мониторинга, геоинформационный интернет-сервер, веб-картография, геопространственные данные.

Оперативный анализ состояния дел в регионе и мониторинг основных социально-экономических, финансово-бюджетных и прочих показателей деятельности в разрезе муниципальных образований являются сегодня одними из приоритетных задач органов власти и управления. Сложность их решения связана прежде всего с рядом организационных и административных проблем, ведомственной разобщенностью. Другой важной проблемой является техническая – отсутствие согласованной информационной политики межведомственного информационного взаимодействия, технологических стандартов на данные и проч. Несмотря на отдельные успешные отраслевые IT-проекты, достаточно типичной продолжает оставаться ситуация, когда сбор данных осуществляется в «телефонном» или «бумажном» формате. Оценивать регион в целом и проводить сравнительный анализ его муниципальных образований в этой ситуации сложно и

малоэффективно. И даже при наличии электронных документов разнообразие их форм и форматов создает определенную сложность при выработке решений [1].

Представленная в данной работе модель геоинформационной аналитической интернет-системы рассматривается в контексте указанной проблемы. Она ориентирована на пользователей, которые не являются профессионалами в ГИС, но владеют базовыми навыками использования Интернета. Основное внимание уделяется современным средствам визуализации данных в рамках веб-интерфейса. Отличительной особенностью является возможность формирования оценок состояния территорий на базе основных показателей в полуавтоматическом режиме [2].

С технической точки зрения рассматриваемая здесь модель системы – это веб-портал, содержащий коллекцию типизированных компонентов-фреймов, представ-

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 09-07-98002-р_сибирь_a) и гранта Президента РФ для ведущих научных школ (код проекта НШ-3431.2008.9).