

# КЛАССИФИКАЦИЯ МЕТОДОВ ПАРАЛЛЕЛЬНОГО МОДЕЛИРОВАНИЯ В ЗАДАЧАХ ДИАГНОСТИКИ ЦИФРОВЫХ УСТРОЙСТВ

УДК 004.3:681.518

**ИВАНОВ Дмитрий Евгеньевич**

д.т.н., доцент, с.н.с. отдела теории управляющих систем ИПММ НАН Украины.

**Научные интересы:** техническая диагностика цифровых систем, эволюционные вычисления в технической диагностике, генетические алгоритмы, параллельные вычисления.

**e-mail:** ivanov@iamm.ac.donetsk.ua

## ВВЕДЕНИЕ ПОСТАНОВКА ЗАДАЧИ

Моделирование поведения разрабатываемого цифрового устройства (ЦУ) – одна из важнейших задач жизненного цикла проектирования [1]. Моделирование поведения ЦУ имеет два основных назначения. Во первых, данные методы имеют самостоятельную ценность, позволяя с выбранной адекватностью моделировать поведение разрабатываемого устройства, в том числе в присутствии с неисправностей, с целью валидации поведения заданной спецификации. Во вторых, они применяются для оценки строящихся диагностических последовательностей широкого класса в методах, которые основаны на моделировании. Примером таких интенсивно развивающихся в настоящее время методов являются эволюционные методы построения тестов [2]. Моделирование поведения ЦУ может осуществляться на различных уровнях представления [1], что делает задачу их обзора крайне затруднительной. В данной статье под моделированием ЦУ будем понимать данную задачу для логического уровня представления.

Большой размер и сложность современных проектируемых ЦУ требует больших временных ресурсов вычислительной системы (ВС) для их моделирования. Так, в 1998 моделирование на логическом уровне устройства, включающего в себя миллион вентилях, на входной последовательности длиной 1000 входных наборов оценивалось в 24 часа машинного времени

для рабочей станции DEC [3]. При этом согласно Международной Дорожной Карте [4] число транзисторов в SOC-системах потребительского уровня увеличится в 17 раз к 2024 году, а длина входных последовательностей для тестирования наиболее часто используемых на логическом уровне одиночных константных неисправностей (ОКН) увеличится с нынешних 30 тысяч входных векторов в 16 раз к указанному времени.

Одним из очевидных способов повышения быстродействия методов моделирования является разработка их параллельных версий. К настоящему моменту времени разработано множество методов параллельного моделирования как для исправных ЦУ, так и для ЦУ с неисправностями. Это объясняется разнообразием типов параллельных ВС, а также применяемых подходов. Целью данной статьи является обзор таких методов моделирования ЦУ, на основании которого предлагается их классификация. Это позволит разработчикам ЦУ решать две основные задачи. Во первых, появится возможность выбирать соответствующие базовые параллельные методы, подходы их конструирования либо эвристики оптимизации для построения новых методов решения конкретных задач, возникающих при проектировании ЦУ. При этом такие методы могут быть реализованы для работы на современных параллельных ВС. Во вторых, это обеспечит адаптацию существующих методов для работы на доступных разработчику параллельных аппаратных платформах. В такой постановке задача не изучалась.

## ПАРАЛЛЕЛЬНОЕ МОДЕЛИРОВАНИЕ ИСПРАВНЫХ ЦУ

Одна из центральных задач в проектировании ЦУ – валидация синтезированного устройства на соответствие заданной спецификации, которая реализуется с помощью методов моделирования исправных ЦУ. Поскольку здесь выполняется моделирование одного устройства, то загрузка нескольких вычислительных процессоров достигается путём разбиения данного ЦУ на части с дальнейшим моделированием каждой под-схемы на своём процессоре/узле. Такой подход является основным и описан, например, в [5-7]. Далее будет отмечено, что такой же подход применим и для моделирования ЦУ с неисправностями. Основным преимуществом подхода является то, что резко уменьшаются требования к затратам памяти, в которой хранится описание схемы. Поэтому разработка методов моделирования с разбиением схемы была характерна для начала эры параллельных ВС, когда каждый узел имел ограниченную память, однако их совокупность позволяла выполнять моделирование поведения устройства в целом [6]. Несмотря на такое замечание, подход с разбиением схемы применяется и в настоящее время [8]. Например, в [7] описывается метод данного класса, который разработан для графических ускорителей с большим числом потоковых процессоров.

Основными недостатками подхода являются:

- относительно небольшой размер обрабатываемых подсхем, что ведёт к увеличению данных, передаваемых по линиям связей;
- необходимость взаимодействия отдельных задач друг с другом в процессе работы, что требует написания протоколов взаимодействия, а также ведёт к сложности в реализации.

Для оптимизации процесса взаимодействия узлов в ВС разрабатываются два основных подхода, включающие «консервативные» и «оптимистические» протоколы. Оптимистические методы синхронизации допускают нарушение очередности выполнения событий в подсхемах, но в дальнейшем они обнаруживают и исправляют их. Данный подход более распространён и имеет два важных преимущества в сравнении с консервативными методами. Во-первых, оптимистические протоколы могут использовать большую степень параллелизма. Во-вторых, консервативный подход обыч-

но полагается на применение некоторой информации, для того чтобы определить, какие события являются безопасными. Оптимистические алгоритмы работают без такой информации. С другой стороны, оптимистические методы могут потребовать больше накладных расходов как для вычислений, так и для передачи данных, чем консервативные подходы, что приводит к снижению производительности.

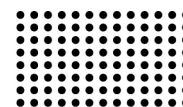
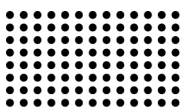
Одним из наиболее известных оптимистических алгоритмов является Time Warp [9], который также является основой для многочисленных улучшений. Принцип его работы заключается в выполнении событий без гарантии соблюдения последовательного порядка. Если возникает ситуация нарушения естественного порядка временных меток событий, протокол производит откат к необходимому состоянию, используя механизм анти-сообщений. Известны также работы в данном направлении отечественных авторов [10].

В целом, данные методы развиваются в направлении, позволяющем сбалансировать нагрузку узлов в системе при сдерживании объёма передаваемой информации [8, 11].

## ПАРАЛЛЕЛЬНОЕ МОДЕЛИРОВАНИЕ ЦУ С НЕИСПРАВНОСТЯМИ

Второй важной задачей является моделирование ЦУ с неисправностями. Задачи данного рода возникают в двух случаях. Традиционно, такое моделирование используется для определения качества построенной диагностической последовательности в соответствии с заданной метрикой, например, полноты теста. Второе применение связано с развитием методов построения идентифицирующих последовательностей (ИдП), основанных на моделировании. Здесь происходит оценка потенциальных решений-последовательностей. К данному направлению относятся различные эволюционные методы построения ИдП [12-13, 2] – генетические алгоритмы, моделирование отжига и т.п.

Следует отметить, что задача моделирования с неисправностями является более сложной в сравнении с моделированием исправных ЦУ. Это связано с размером списка неисправностей, которая прямо пропорциональна размерности обрабатываемого устройства. При этом, например, для одиночных неисправностей отдельно необходимо выполнять модели-



рование ЦУ в присутствии каждой неисправности из списка. В случае кратных неисправностей задача ещё более усложняется. Именно поэтому для современных ЦУ процесс моделирования может продолжаться до нескольких десятков часов, что является наибольшим недостатком указанных методов. Поэтому задача построения быстрых алгоритмов моделирования с неисправностями на сегодняшний день остаётся актуальной. С развитием современных параллельных ВС различной аппаратной конфигурации задача продолжает быть интересной для исследователей.

Техники, ускоряющие моделирование ЦУ с неисправностями, следует разделить на два класса. К первому относится параллельное по разрядам машинного слова моделирование, тогда как ко второму – непосредственно параллельное моделирование, использующее ресурсы нескольких вычислительных процессоров.

### **ПАРАЛЛЕЛЬНОЕ ПО РАЗРЯДАМ МАШИННОГО СЛОВА МОДЕЛИРОВАНИЕ ЦУ С НЕИСПРАВНОСТЯМИ**

Данный подход исторически был развит первым, поскольку параллельность обработки информации в машинном слове является фундаментальным свойством всех современных ЭВМ, которое было сформулировано ещё фон Нейманом в 1945 году. Строго говоря, такой подход не является «параллельным», поскольку не использует вычислительных ресурсов нескольких процессоров. Однако одновременная обработка нескольких устройств, позволяющая существенно повысить быстродействие, а также очень широкое распространение позволяют нам включить данную группу методов в разрабатываемую иерархию.

Параллельные по разрядам машинного слова метод моделирования ЦУ с неисправностями используют технику параллельного одиночного распространения влияния неисправностей. Наиболее известным является метод PROOFS [14], который явился основой для многочисленных улучшений [15], в том числе и от авторов данной работы [16].

Применяемая техника заключается в том, что исправное ЦУ моделируется в первом разряде машинного слова, остальные  $n - 1$  разрядов моделируют  $n - 1$  неисправных ЦУ, соответственно, где  $n$  - число разрядов инструментальной ЭВМ. Применение такого под-

хода возможно поскольку структуры всех параллельно моделируемых устройств одинаковы. Для неисправностей, которые моделируются в одном машинном слове параллельно (группе), большинство событий моделирования совпадают. Таким образом, общее число событий моделирования для группы неисправностей растёт существенно медленнее, чем число неисправностей в группе.

Развитие методов данного класса связано с применением оптимизационных эвристик, уменьшающих число событий моделирования, и техниками внесения влияния неисправностей. Группировка неисправностей, порождающих одинаковые события, резко повышает быстродействие методов. В [17] предлагается динамическая пересортировка потенциально проверяемых и непроверяемых неисправностей, в [15] предлагается отдельное моделирование гиперактивных неисправностей.

Первоначально внесение влияния неисправностей, а также сравнение поведения неисправных ЦУ с исправным устройством выполнялось с помощью битовых масок. Для уменьшения времени внесения влияния неисправностей в [14] используется техника дополнительных вентилях. При этом для сохранения структуры устройств во всех разрядах такие вентили добавляются во все моделируемые устройства. В [18] предложена техника фиктивных вентилях, позволяющая избежать как проверки необходимости внесения неисправности перед моделированием каждого вентиля, так и добавления в устройство дополнительных вентилях, что существенно повышает скорость работы. В [16] данная техника распространена на параллельное по разрядам моделирование ЦУ с неисправностями.

### **МОДЕЛИРОВАНИЕ ЦУ С НЕИСПРАВНОСТЯМИ НА ПАРАЛЛЕЛЬНЫХ ВС**

Разработка таких методов стала возможной с развитием параллельных ВС [19], причём первые работы относятся ещё к концу 80-х годов прошлого столетия [20]. На наш взгляд, наиболее полно вопрос построения параллельного алгоритма моделирования схем с неисправностью рассмотрен в [21], где предложена параллельная модификация алгоритма PROOFS. Цель данного алгоритма – быстрое определение диагностических свойств представленной входной последовательности. Отметим, что в случае, когда необходимо восстанавли-

вать поведение схемы в присутствии одной либо нескольких неисправностей, данный подход не применим, поскольку неисправность исключается из рассмотрения в тот момент времени, когда она обнаружена в первый раз. Это замечание относится ко всем методам, использующим технику параллельного одиночного распространения неисправностей.

К настоящему времени выделились три основные схемы распараллеливания алгоритмов моделирования с неисправностями.

1) Разбиение списка неисправностей [19, 22-23]. При данном подходе полный список неисправностей  $F$  разбивается на несколько подсписков  $F_1, F_2, \dots, F_n$ , каждый из которых передаётся на отдельный процессор системы, где и выполняется его анализ путём моделирования на заданной входной последовательности. При такой схеме каждый узел вычислительной среды должен иметь свою копию описания схемы и теста, что требует организации соответствующих процедур обмена данными. В настоящее время данный метод получил наибольшее распространение и считается, что он обладает хорошей масштабируемостью при росте числа процессоров. Авторы также предлагали алгоритм данного рода как для кластерных систем с распределённой памятью [24], так и для многоядерных рабочих станций с общей памятью [25].

2) Разбиение схемы [5, 26]. Идеологически данный подход полностью аналогичен применяемому для параллельного моделирования исправных ЦУ: схема разбивается на несколько подсхем, каждая из которых моделируется на отдельном узле вычислительной системы.

3) Разбиение теста [27] заключается в том, что входная последовательность  $T$  разбивается на ряд подпоследовательностей  $T_1, T_2, \dots, T_m$ , с которыми и формируются подзадачи для вычислительных узлов. Процессы также должны взаимодействовать, чтобы информировать друг друга о том, какие неисправности обнаружены к текущему моменту времени. В случае, если в заданной входной последовательности не известны такты времени, в которые начинаются подпоследовательности (что случается довольно часто), данный метод не может быть применён.

Подходы с разбиением списка неисправностей и разбиением теста дополнительно могут использовать параллельное по разрядам машинного слова моделирование для каждого вычислительного узла.

В настоящее время развитие параллельных методов происходит за счёт их адаптации к многоядерным рабочим станциям, а также к многопоточным графическим ускорителям.

Первоначально многоядерные процессоры содержали 2 вычислительных ядра. Это позволило стать им основой целого ряда мощных многопроцессорных ВС. Сейчас коммерчески доступными являются рабочие станции, в которых содержится 1-2 процессора, каждый из которых содержит 6-8 вычислительных ядер [28]. Между тем практически отсутствуют исследования по эффективности параллельных методов моделирования именно для таких систем. Можно отметить публикацию [29], но здесь исследуется моделирование на системном уровне представления. В отечественной литературе также данное направление представлено единичными публикациями [30]. Часто зарубежные авторы не производят различия между параллельными ВС и многоядерными рабочими станциями. Между тем очевидно, что такие системы обладают своими особенностями в сравнении с распределёнными ВС общего назначения, что должно учитываться при разработке новых методов. Интересной в этом смысле является работа авторов [31]. В данной работе предложен алгоритм моделирования с неисправностями для многоядерной ВС с общей памятью, который не вписывается в приведённую классификацию. Он заключается в том, что распараллеливанию подвергаются динамически сформированные группы неисправностей для каждого входного набора. При этом неисправности в каждой такой группе дополнительно моделируются параллельно по разрядам машинного слова. Также на основании экспериментов показано, что традиционная техника моделирования исправного ЦУ в отдельном разряде машинного слова требует модификации. Оказалось, что эффективнее моделировать поведение исправного ЦУ не в отдельном разряде машинного слова каждой группы неисправностей, а в отдельном вычислительном потоке. Поведение ЦУ в нём является эталонным для всех групп неисправностей текущей итерации.

В последнее время в связи с выходом графических ускорителей с большим числом потоковых процессоров (GPU) оформилось ещё одно направление в параллельном моделировании, которое заключается в разработке методов моделирования для таких GPU. Алгоритмы данного типа предложены в [32-35]. Они могут являться специальной версией для указанных GPU, либо использовать новые подходы и эвристики. Например, в [32] предлагается понятие макро-вентилей (macro-gates), моделирование которых на

узлах производится в только случае их активации (изменения сигнала на его входах). Такой подход переносит понятие событийного моделирования на уровень макро-вентилей. В целом, данные алгоритмы по производительности существенно превосходят параллельные алгоритмы для остальных систем, в основном, за счёт очень большого числа параллельных процессоров. Однако мы не будем выделять алгоритмы данного рода в отдельную группу, поскольку их идеология вписывается в классификацию выше.

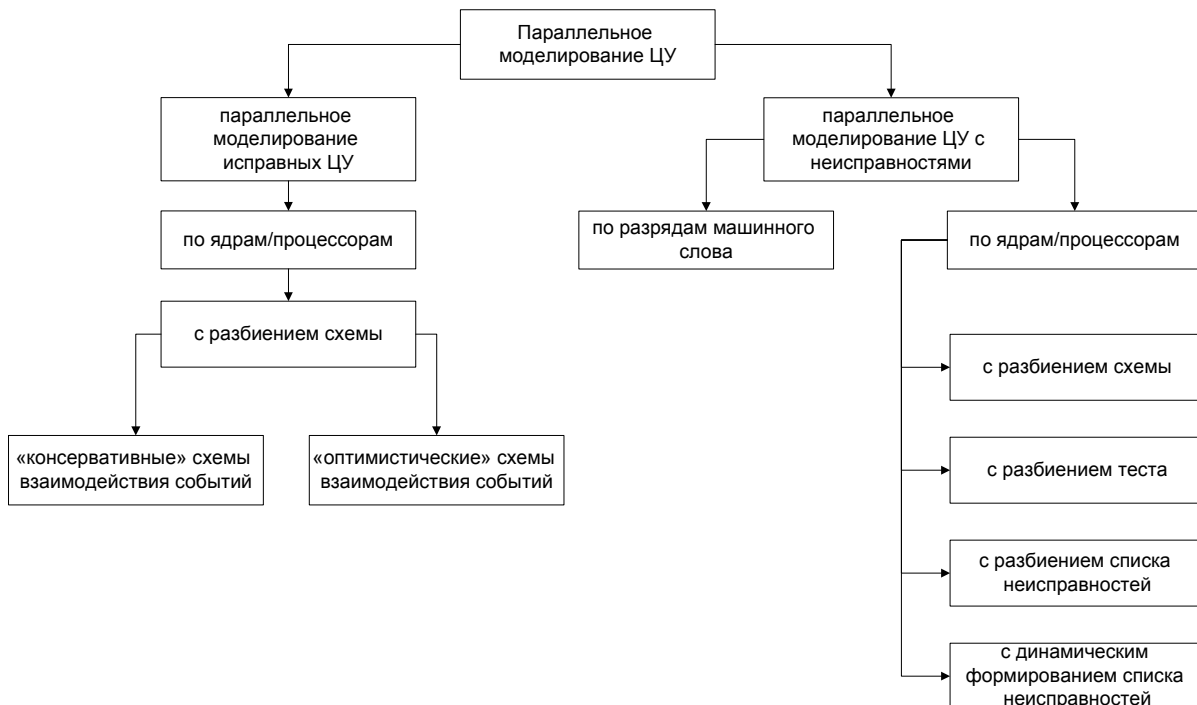


Рисунок 1 – Классификация методов параллельного моделирования ЦУ

## ИЕРАРХИЯ МЕТОДОВ ПАРАЛЛЕЛЬНОГО МОДЕЛИРОВАНИЯ ЦУ

Рассмотренные в обзоре выше методы параллельного моделирования охватывают все основные предложенные к настоящему моменту подходы построения параллельных методов моделирования для различных параллельных ВС. Исходя из проведённого анализа, классификация таких методов в виде укрупнённой древовидной иерархии может быть представлена как на рис. 1.

На данной диаграмме мы не разделяли методы моделирования по назначению для параллельных ВС с общей и разделяемой памятью, поскольку общие подходы применяются для обоих классов систем. Однако отдельно был выделен метод параллельного моделирования ЦУ с неисправностями с динамическим разбиением списка неисправностей [31]. Он апробирован только на сильнопараллельной ВС с общей памятью, для которой он, собственно, и разрабатывался. Экспериментов с данным методом для параллельных систем

с разделённой памятью к настоящему времени не известно.

Целью построения данной классификации является предоставить разработчику комплекс подходов «конструирования» параллельных методов моделирования, что должно помочь при разработке новых таких методов в процессе решения задач проектирования ЦУ.

Разработка новых методов моделирования происходит в два этапа. На первом этапе в соответствии с целевой аппаратной платформой происходит декомпозиция уровней метода сверху вниз. При этом для каждого аппаратного уровня выбирается свой базовый метод моделирования. На втором этапе происходит синтез общего метода, который выполняется в обратном порядке (снизу вверх) на основании алгоритмической реализации методов каждого из уровней.

Покажем применение такого подхода на примере. Пусть необходимо разработать метод параллельного моделирования ЦУ с неисправностями для параллельной ВС с распределённой памятью. В данном случае в зависимости от требуемых условий на верхнем уровне может быть выбран метод с разбиением списка неисправностей. Каждый из подсписков проектируется на вычислительный узел. Далее, если узел содержит слабопараллельный процессор (2-4 вычислительных ядра), то снова применяется подход с разбиением списка неисправностей, при этом каждый подспик проектируется на своё вычислительное ядро. Если же, к примеру, каждый узел содержит процессор с 12-16 ядрами, можно применить подход с динамическим разбиением списка неисправностей. На самом нижнем уровне может быть применён метод параллельного по разрядам

машинного слова моделирования, который отображается на машинное слово. Таким образом видно, что разрабатываемый метод иерархически разбит на три уровня. Дальнейшая композиция реализации метода (его синтез) основывается на алгоритмической реализации каждого из уровней. Также при дальнейшем уточнении метода и его алгоритмической реализации полезным является применение многочисленных оптимизационных эвристик, что дополнительно позволит ускорить работу программной реализации.

### ЗАКЛЮЧЕНИЕ

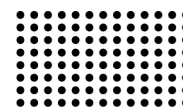
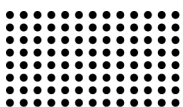
В работе проведён обзор параллельных методов моделирования ЦУ (исправных и с неисправностями) и выполнен анализ их применения. На основании данного обзора предложена классификация данных методов, представленная в виде укрупнённой древовидной диаграммы.

На основании данной диаграммы предложен двухэтапный подход синтеза новых методов параллельного моделирования. На первом этапе происходит иерархическая декомпозиция метода в соответствии со структурой аппаратуры целевой параллельной ВС. Второй этап заключается в синтезе метода в целом на основании реализации метода каждого из уровней.

Применение такого подхода позволяет как выполнять адаптацию существующих, так и разрабатывать новые методы моделирования, которые возникают при решении задач проектирования и диагностирования ЦУ, для параллельных ВС различной архитектуры.

### ЛИТЕРАТУРА:

1. Skobtsov Yu.A. Logicheskoe modelirovanie i testirovanie cifrovyykh ustrojstv /Yu.A. Skobcov, V.Yu. Skobcov. – Doneck:IPMM NANU, DonNTU, 2005. – 436 s.
2. Ivanov D.E. Geneticheskie algoritmy postroeniya vkhodnykh identifiatsionnykh posledovatel'nostey cifrovyykh ustrojstv /D.E. Ivanov. – Doneck, 2012. – 240 s.
3. Kim H.K. Parallel logic simulation of digital circuits /Hong Kyu Kim //A dissertation for the degree of doctor of philosophy, Wright State University, USA, 1998. – 100 p.
4. ITRS 2010 technology roadmap [e'lektronnyj resurs] //http://www.itrs.net/Links/2010ITRS/Home2010.htm/
5. Ghosh S. A distributed algorithm for fault simulation of combinatorial and asynchronous sequential digital designs, utilizing circuit partitioning, on loosely coupled parallel processors /S. Ghosh //Microelectronic Reliability. – 1995. – №35(6). – P.947-967.
6. Muller-Thuns R.B. Abraham Portable parallel logic and fault simulation /R.B. Muller-Thuns, D.G. Saab, R.F. Damiano, J.A. Abraham //Digest of paper, Int. Conference on Computer Aided Design /Santa Clara, USA. – 1989. – P.506-509.
7. Subbaraj P. Circuit Partitioning Problem using Graphical Processing Units /P. Subbaraj, P. Sivakumar, S. Nandhanam //Journal of Computer Science. – 2012. – №8 (5). – P.705-710.
8. Meraji S. On the Scalability and Dynamic Load-Balancing of Time Warp /Sina Meraji, Wei Zhang, Carl Troppe //IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2010. – Vol.29, №9. – Pp.1368-1380.



9. Palaniswamy A.C. Parameterized Time Warp (PTW): An Integrated Adaptive Solution to Optimistic PDES /Avinash C. Palaniswamy, Philip A. Wilsey //Journal of Parallel and Distributed Computing. – 1996. – Vol.37, Issue 2. – Pp.134-145.
10. Ladyzhenskij, Yu.V. Programmnyaya sistema dlya issledovaniya protokolov sinxronizatsii pri raspredelyennom sobytijnom logicheskom modelirovanii /Yu.V. Ladyzhenskij, Yu.V. Popov //Naukovi praci Donec'kogo nacional'nogo texnichnogo universitetu, Seriya "Obchislyval'na tekhnika ta avtomatizatsiya". –2004. – №74. – S.201-209.
11. Overinder B.J. Distributed Event-driven Simulation. Scheduling Strategies and Resource Management. Scheduling Strategies and Resource Management //A dissertation for the degree of doctor of philosophy, Universiteit van Amsterdam, 2000. – 228 p.
12. Srinivas M. A simulation-based test generation scheme using genetic algorithms /M. Srinivas, L.M. Patnaik //Proc. Int. Conf. VLSI Design. – 1993. – P.132-135.
13. Como F. Experiences in the use of evolutionary techniques for testing digital circuits /F. Como, M. Sonza Reorda, M. Rebaudengo //Proc. of Conf. Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA. – 1998. – P.128-139.
14. Niemann T.M. PROOFS: A Fast, Memory-Efficient Sequential Circuits Fault Simulator /T.M. Niemann, W.-T. Cheng, J.H. Patel //IEEE Trans. CAD. – 1992. – V.11. – №2. – P.198-207.
15. Kung C.P. HyHope: A Fast Fault Simulator with Efficient Simulation of Hypertrophic Faults /C.P. Kung, C.S. Lin //Proc. of International Test Conference. – 1994. – P.714-718.
16. Ivanov D.E. Parallelnoe modelirovanie neispravnostej dlya posledovatel'nostnyx sxem /D.E. Ivanov, Yu.A. Skobcov //Iskusstvennyj intellekt. – 1999. – №1. – S.44-50.
17. Graham C.R. Dynamic fault grouping for PROOFS: a win for large sequential circuits /C.R. Graham, E.M. Rudnick, J.H. Patel //Proceedings of Tenth International Conference on VLSI Design, 1997. – Pp.542-544.
18. Lee H.K. New Method of Improving Parallel Fault Simulation in Synchronous Sequential Circuits. /H.K. Lee, D.S. Ha //Proc. Int. Conf. On Computer-Aided Design. – 1993. – P.10-17.
19. Mueller-Thuns R.B. VLSI Logic and Fault Simulation on General-Purpose Parallel Computers /R.B. Mueller-Thuns, D.G. Saab, R.F. Damiano, J.A. Abraham //IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1993. – №12 (3). – P.446-460.
20. Duba P.A. Fault Simulation in a Distributed Environment /P.A. Duba, R.K. Roy, J.A. Abraham, W.A. Rogers //Proceedings of 25th Design Automation Conference. – 1988. – P.686-691.
21. Parker S. A parallel algorithm for fault simulation based on PROOFS /S. Parker, P. Banerjee, J. Patel //Proc. IEEE Int. Conf. Computer Design. – 1995. – P.616-621.
22. Krishnaswamy D. Asynchronous parallel algorithms for test set partitioned fault simulation /D. Krishnaswamy, P. Banerjee, E.M. Rudnick, J.H. Patel //ACM SIGSIM Simulation Digest. – 1997. – Volume 27, Issue 1. – P.30-37.
23. Markas T. On distributed fault simulation /T. Markas, M. Royals, N. Kanopoulos //IEEE Computer. – 1990. – Vol.7. – P.40-52.
24. Ivanov D.E. Raspredelyonnoe parallelnoe modelirovanie cifrovyyx sxem s neispravnostyami /D.E. Ivanov, Yu.A. Skobcov, E.I'-Xatib A.I. //Naukovi praci Donetskoho natsionalnogo tekhnichnogo universitetu. Seriya: "Obchislyval'na tekhnika ta avtomatyzatsiia". Vypusk 107. – 2006. – P.128-134.
25. Ivanov D.E. Parallel fault simulation on multi-core processors /D.E. Ivanov //Radioelektronni i kompiuterni systemy. – 2009. – №6(40). – C.109-112.
26. Subbaraj P. Circuit Partitioning Problem using Graphical Processing Units /P. Subbaraj, P. Sivakumar, S. Nandhanam //Journal of Computer Science. – 2012. – №8 (5). – P.705-710.
27. Ravikumar C.P. Distributed Fault Simulation Algorithms on Parallel Virtual Machine /C.P. Ravikumar, V. Jain, A. Dod //VLSI Design. – 2001. – Volume 12, Issue 1. – P.81-99.
28. Intel® Software Insight. Multi-core Capability /R. Wirt. – USA: Intel Corporation, 2005, July. – 11 p.
29. Dömer R. Multi-core parallel simulation of system-level description languages /R. Dömer, W. Chen, X. Han //Proceedings of the 16th Asia and South Pacific Design Automation Conference, 2011. – P.311-316.
30. Hahanov V. Parallel Logic Simulation using Multi-Core Workstations /V. Hahanov, V. Obrizan, A. Gavryushenko, S. Mikhtonyuk //The Experience of Designing and Applications of CAD Systems in Microelectronics, 2007. CADSM '07. 9th International Conference. – 2007. – P.256-257.
31. Ivanov D.E. Parallelnyj algoritm modelirovaniya cifrovyyx sxem s neispravnostyami dlya mnogoyademyx sistem s obshhej pamyat'yu /D.E. Ivanov //Elektronnoe modelirovanie. – 2011. – T.33, №2. – S.93-106.
32. Chatterjee D. High-performance gate-level simulation with GP-GPUs /D. Chatterjee, A. DeOrio, V. Bertacco //In Proceedings of DATE, 2009. – P.1332-1337.
33. Gulati K. Towards acceleration of fault simulation using graphics processing units /K. Gulati, S. Khatri //Proceedings of the 45th annual Design Automation Conference, DAC '08. – 2008. – P.822-827.
34. Perinkulam A. Logic simulation using graphics processors /A. Perinkulam, S. Kundu //In Proc. ITSW, 2007. (v e'lektronnom vide)
35. Kochte M.A. Efficient Fault Simulation on Many-Core Processors /M.A. Kochte, M. Schaal, H.-J. Wunderlich, C.G. Zoellin //Proceedings of the 47th Design Automation Conference ACM, New York, NY, USA. – 2010. – P.380-385.

**Рецензент:** д.т.н., проф. Скобцов Ю.А.,  
Донецкий национальный технический университет, Донецк.