

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ,  
МОЛОДЕЖИ И СПОРТА УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ  
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN0135-1710

# **АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ**

**Всеукраинский межведомственный  
научно-технический сборник**

**Основан в 1965 г.**

**Выпуск 159**

Харьков  
2012

В сборнике представлены результаты исследований, касающихся компьютерной инженерии, управления, технической диагностики, автоматизации проектирования, оптимизированного использования компьютерных сетей и создания интеллектуальных экспертных систем. Предложены новые подходы, алгоритмы и их программная реализация в области автоматического управления сложными системами, оригинальные информационные технологии в науке, образовании, медицине.

Для преподавателей университетов, научных работников, специалистов, аспирантов.

У збірнику наведено результати досліджень, що стосуються комп'ютерної інженерії, управління, технічної діагностики, автоматизації проектування, оптимізованого використання комп'ютерних мереж і створення інтелектуальних експертних систем. Запропоновано нові підходи, алгоритми та їх програмна реалізація в області автоматичного управління складними системами, оригінальні інформаційні технології в науці, освіті, медицині.

Для викладачів університетів, науковців, фахівців, аспірантів.

**Редакционная коллегия:**

*В.В. Семенец, д-р техн. наук, проф. (гл. ред.); М.Ф. Бондаренко, д-р техн. наук, проф.; И.Д. Горбенко, д-р техн. наук, проф.; Е.П. Пулятин, д-р техн. наук, проф.; В.П. Тарасенко, д-р техн. наук, проф.; Г.И. Загарий, д-р техн. наук, проф.; Г.Ф. Кривуля, д-р техн. наук, проф.; Чумаченко С.В., д-р техн. наук, проф.; В.А. Филатов, д-р техн. наук, проф.; Е.В. Бодянский, д-р техн. наук, проф.; Э.Г. Петров, д-р техн. наук, проф.; В.Ф. Шостак, д-р техн. наук, проф.; В.М. Левыкин, д-р техн. наук, проф.; Е.И. Литвинова, д-р техн. наук, проф.; В.И. Хаханов, д-р техн. наук, проф. (отв. ред.).*

Свидетельство о государственной регистрации  
печатного средства массовой информации

КВ № 12073-944ПР от 07.12.2006 г.

*Адрес редакционной коллегии:* Украина, 61166, Харьков, просп. Ленина, 14, Харьковский национальный университет радиоэлектроники, комн. 321, тел. 70-21-326

© Харківський національний університет  
радіоелектроніки, 2012

## СОДЕРЖАНИЕ

<b>ЛИСИЦКАЯ И.В., НАСТЕНКО А.А., ЛИСИЦКИЙ К.Е.</b> БОЛЬШИЕ ШИФРЫ – СЛУЧАЙНЫЕ ПОДСТАНОВКИ. СРАВНЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ И ЛИНЕЙНЫХ СВОЙСТВ ШИФРОВ, ПРЕДСТАВЛЕННЫХ НА УКРАИНСКИЙ КОНКУРС, И ИХ УМЕНЬШЕННЫХ МОДЕЛЕЙ .....	4
<b>МУРАД АЛИ АББАС, БАГХДАДИ АММАР АВНИ АББАС, ХАХАНОВ В.И., ЛИТВИНОВА Е.И., ДАХИРИ ФАРИД</b> ТЕХНОЛОГИИ ВОССТАНОВЛЕНИЯ РАБОТОСПОСОБНОСТИ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ НА КРИСТАЛЛАХ .....	10
<b>ЗЕМЛЯК А.М., МАРКИНА Т.М.</b> ХАРАКТЕРИСТИКИ РАЗЛИЧНЫХ СТРАТЕГИЙ ПРОЕКТИРОВАНИЯ АНАЛОГОВЫХ ЦЕПЕЙ В РАСШИРЕННОМ БАЗИСЕ .....	23
<b>ЛЕВЫКИН И.В., ЛОГВИНЕНКО Е.В.</b> УСОВЕРШЕНСТВОВАННЫЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ ОПИСАНИЯ ХАРАКТЕРИСТИК ОПЕРАЦИИ ЗАКАЗА И ОБОРУДОВАНИЯ ПОЛИГРАФИЧЕСКОГО ПРЕДПРИЯТИЯ .....	30
<b>ФЕДОРОВА Т.Н.</b> О ПОДХОДЕ К ПОСТРОЕНИЮ ЦЕПОЧЕК ЛЕКСИЧЕСКИХ ЕДИНИЦ УКРАИНСКОГО ЯЗЫКА В ЛЕКСИКОГРАФИЧЕСКОЙ СИСТЕМЕ ЭЛЕКТРОННОГО ТОЛКОВОГО СЛОВАРЯ .....	33
<b>ФИЛАТОВ В.А., АРТЮХ Р.В.</b> МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ВАРИАНТОВ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ В БАЗЕ ПРЕЦЕДЕНТОВ .....	40
<b>КОЖУХІВСЬКА О.А.</b> МОДЕЛЮВАННЯ ФІНАНСОВИХ РИЗИКІВ З ВИКОРИСТАННЯМ ЙМОВІРНІСНОГО ПІДХОДУ .....	46
<b>ГЕРАСИМЕНКО К.Е.</b> МЕТОД ПОВЫШЕНИЯ КОНТРОЛЕПРИГОДНОСТИ КРИТИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ АЭС.....	53
<b>ПОЧАНСКИЙ О.М.</b> ЭКСПЕРТНАЯ СИСТЕМА СЕМАНТИЧЕСКОГО ПОИСКА РЕЛЕВАНТНЫХ ДАННЫХ И ФОРМИРОВАНИЯ АДАПТИВНЫХ WEB-СТРАНИЦ .....	57
<b>ГАЛУШКА І.М., ЗАВГОРОДНІЙ В.В., СОЛОШИЧ С.М., ЩЕРБАК С.С.</b> УДОСКОНАЛЕННЯ ТЕХНОЛОГІЙ ДОСТУПУ ТА ОБРОБКИ ПОВ'ЯЗАНИХ ДАНИХ СЕМАНТИЧНИХ ДОДАТКІВ LINKEDDATA .....	67
<b>ОКСАНИЧ А.П., ПРИТЧИН С.Э., МАЛЁВАНЫЙ В.В.</b> РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ И УСТРОЙСТВА АВТОМАТИЧЕСКОГО КОНТРОЛЯ И ПОДДЕРЖАНИЯ ДИАМЕТРА СЛИТКОВ ГЕРМАНИЯ, ВЫРАЩИВАЕМЫХ ПО МЕТОДУ ЧОХРАЛЬСКОГО ....	73
<b>РЕФЕРАТЫ.....</b>	80
<b>ПРАВИЛА</b> ОФОРМЛЕНИЯ СТАТЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО СБОРНИКА .....	84

Кузнецов А.А. Симметричный криптографический алгоритм ADE (Algorithm of Dynamic Encryption). / А.А. Кузнецов, Р.В. Сергиенко, А.А. Наушко // Прикладная радиоэлектроника. Харьков: ХТУРЭ. 2007. Т. 6, №2. С. 241-249. **11.** Krzysztof Chmiel. On Differential and Linear Approximation of S-box Functions / Biometrics, Computer Security Systems and Artificial Intelligence Applications. / Edited by Khalid Saeed, Jerzy Pejas and Romuald Mosdorf // Poland, Springer. 2006. P. 111-120. **12.** Олейников Р.В. Дифференциальные свойства подстановок / Р.В. Олейников, О.И. Олешко, К.Е. Лисицкий, А.Д. Тевяшев // Прикладная радиоэлектроника. 2010. Т.9, № 3. С. 326-333. **13.** Долгов В.И. Свойства таблиц линейных аппроксимаций случайных подстановок / В.И. Долгов, И.В. Лисицкая, О.И. Олешко // Прикладная радиоэлектроника. Харьков: ХНУРЭ. 2010. Т. 9, № 3. С. 334-340.

Поступила в редколлегию 23.06.2012

**Лисицкая Ирина Викторовна**, д-р техн. наук, доцент кафедры безопасности информационных технологий ХНУРЭ. Научные интересы: криптография, методы криптоанализа. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел.: +38 (057) 702-14-25, E-mail: ai@kture.kharkov.ua.

**Настенко Андрей Александрович**, аспирант кафедры безопасности информационных технологий ХНУРЭ. Научных интересов: криптография, методы криптоанализа. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

**Лисицкий Константин Евгеньевич**, студент ХНУРЭ. Научные интересы: криптография, методы криптоанализа. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

---

УДК 658.512.011:681.326:519.713

*МУРАД АЛИ АББАС, БАГХДАДИ АММАР АВНИ АББАС, ХАХАНОВ В.И.,  
ЛИТВИНОВА Е.И., ДАХИРИ ФАРИД*

## **ТЕХНОЛОГИИ ВОССТАНОВЛЕНИЯ РАБОТОСПОСОБНОСТИ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ НА КРИСТАЛЛАХ**

---

Проводится обзор мультипроцессорных систем на кристаллах (MPSoC) и технологий восстановления их работоспособности. Описываются архитектуры MPSoC, базовая архитектура самовосстановления, метод обнаружения и исправления ошибок в программируемых логических матрицах FPGA.

Цель исследования – аналитический обзор технологий тестирования, ремонта и диагностирования цифровых систем на кристаллах при их проектировании и верификации, ориентированный на существенное увеличение выхода годной продукции и уменьшение времени ее выхода на рынок микроэлектроники.

Для достижения цели необходимо решить задачи, связанные с обзором технологий встроенного ремонта цифровых систем на кристаллах, инфраструктуры сервисного обслуживания и аппаратных решений.

### **1. Мультипроцессорные системы на кристалле**

Создание мультипроцессорных систем-на-кристалле (MPSoC) является важным направлением цифровой электроники [1-6, 18, 22]. Как правило, MPSoC представляет собой систему на кристалле с более чем одним процессором. В современных MPSoC реализованы сложные многопроцессорные модули, которые позволяют решать задачи построения сетей связи (networking), обработки мультимедийной информации и управления в реальном времени (Real Time, RT) при одновременном выполнении ограничений, связанных с потреблением энергии и размерами кристалла.

Мультипроцессор на основе программируемых логических интегральных схем (field-programmable gate array, FPGA) имеет ряд преимуществ перед специализированными интегральными схемами (application-specific integrated circuit, ASIC). А именно: возможности быстрого прототипирования, создания новых архитектур и методов коммуникации. Поэтому мультипроцессоры на основе FPGA, также известные как мультипроцессоры на программируемых кристаллах – Multiprocessor-on-Programmable Chip, MPoPC или «мультипроцессор с программным ядром» или «мягкий мультипроцессор» – используются не только для прототипирования, но и для окончательного проекта тоже. Рост производительности ПЛИС позволяет разработчикам реализовывать мультипроцессорную систему в

одной интегральной схеме FPGA. Компании, занимающиеся производством FPGA, предлагают возможности применения процессоров, специально разработанных для использования в ПЛИС. Современные FPGA оснащены встроенными блоками памяти, периферийных устройств и межсоединений. Сегодня можно реализовать на ПЛИС мультипроцессорную систему, содержащую 80-100 мягких процессоров [1]. Основным недостатком мультипроцессорных систем на основе FPGA является меньшая производительность по сравнению с ASIC. Однако этот недостаток полностью компенсируется их преимуществами: 1) гибкость и реконфигурация; количество мягких процессоров, которые могут быть включены в проект, ограничивается лишь возможностями FPGA; каждый процессор может быть настроен отдельно путем добавления кэш-памяти, периферийных модулей; 2) сокращение времени выхода на рынок; процесс проектирования не включает этап производства интегральной схемы, что значительно сокращает время разработки проекта; 3) уменьшение затрат; в настоящее время FPGA являются относительно дешевыми, что позволяет выполнять проектирование небольшой группой разработчиков; кроме того, если обнаруживаются ошибки проектирования, их легче и дешевле исправить.

Основными производителями интегральных схем, предназначенных для создания мультипроцессорных систем на основе FPGA, являются компании Xilinx и Altera. Xilinx предлагает два основных процессора: мягкий MicroBlaze и жесткий PowerPC. Количество PowerPC ограничено моделью FPGA и не может быть более четырех. Число MicroBlaze ограничивается только логическими ресурсами и может быть до 80-100 единиц в Virtex-5. Для реализации мультипроцессорных систем компанией Xilinx предложен ряд эффективных широко используемых решений: общая шина для связи ядер CoreConnect On-Chip Peripheral Bus (OPB), и 8 Fast Simple Link (FSL) портов в каждом MicroBlaze для эффективного соединения от точки к точке.

Altera предлагает процессоры Nios и Nios II, шину Avalon и средство проектирования EDA SOPC Builder.

Существуют три основные архитектуры мультипроцессорных систем: Master-Slave, конвейерная (Pipeline) и сетевая (Net). Можно комбинировать архитектуры Master-Slave и конвейерную, что широко используется на практике. В сетевой архитектуре нет иерархии процессоров и все они могут передавать информацию друг другу когда это необходимо. Примером описанной архитектуры является симметричный мультипроцессор (SMP), в котором все процессоры идентичны (однородная мультипроцессорная система). В Master-Slave системе один или несколько процессоров являются управляющими, остальные – управляемыми. Конвейерная архитектура целесообразна для потоковых приложений и представляет собой цепь процессоров, каждый из которых сопоставлен определенному этапу конвейера. Решение задач, разделенное во времени, позволяет повысить производительность системы.

Другим важным вопросом является способ организации связи и его физическая реализация. Существует три подхода: 1) точка-точка, где процессоры подключаются напрямую; преимущество – высокая пропускная способность, потому что нет деления канала связи, но при расширении системы данный способ не эффективен; 2) общая шина, традиционный подход, пришедший из однопроцессорных систем, который не является эффективным с точки зрения производительности, поскольку в каждый момент времени шина используется только одним процессором; 3) новый и наиболее перспективный подход – сеть-на-кристалле (Network-on-chip, NoC); при наличии большого количества ядер наилучшим образом сочетает площадь кристалла и производительность. Идея заключается в том, чтобы использовать небольшие маршрутизаторы внутри кристалла для обеспечения связи между всеми ядрами системы с низкими задержками.

Известны два способа передачи информации между процессорами: разделяемая память и передача сообщений. Разделяемая память используется наиболее часто, поскольку FPGA имеет ограниченное количество встроенной памяти и данный способ позволяет ее экономить. В системах с разделяемой памятью имеют место проблемы синхронизации и последовательности работы памяти. Как правило, мультипроцессорные системы с общей памятью используют общую шину. Передача сообщений в основном применяется в конвейерных системах.

Существуют 2 вида мультипроцессорных систем: гетерогенные и однородные. Специализированные проблемно-ориентированные системы обычно являются гетерогенными, поскольку они требуют использования различных видов процессоров (мультимедиа, коммуникации, управление, биоинформатика). Однородные MPSoC построены из одинаковых процессоров. В такой системе количество ядер может быть увеличено без изменения архитектуры. Однородные архитектуры, как правило, используются для систем с параллелизмом данных. Например, в беспроводных базовых станциях, где один и тот же алгоритм применяется для нескольких независимых потоков данных.

В настоящее время используются два основных способа создания многопроцессорных систем на базе FPGA: 1) вручную с применением алгоритма, предлагаемого компаниями по производству FPGA; 2) автоматически с помощью технологии мэппинга [31].

## **2. Восстановление работоспособности систем на кристаллах**

Масштабирование нанoeлектронных схем и устройств достигло сегодня уровня 28 нм и продолжается дальше. Предполагается, что к 2020 году ведущие компании по производству интегральных схем смогут перейти на технологические нормы 10 нм. Увеличение сложности и производительности устройств на кристаллах неизбежно приводит к появлению проблем, связанных с обеспечением их устойчивости и надежности [7-18]. Выигрыш в площади кристалла и потребляемой мощности, полученный от применения нанотехнологий, компенсируется дополнительными накладными расходами на обнаружение неисправностей и исправление ошибок [19-50].

Обнаружение ошибок путем дублирования и исправление их методом «большинства голосов» – широко распространенные технологии, которые могут существенно увеличивать накладные расходы. В последнее время были предложены более сложные методы, где высокий уровень избыточности применяется выборочно только к критическим сигналам, что позволяет значительно уменьшить стоимость устройства. Дублирование, трехкратное резервирование и обнаружение ошибок с помощью специальных кодов в основном ориентированы на перемежающиеся и постоянные неисправности, возникающие в процессе работы. Использование этих методов не целесообразно, если перемежающиеся неисправности возникают на фоне постоянной ошибки. Такие условия могут возникнуть в интегрированных системах, которые должны надежно работать в течение длительного времени. Если две или три копии работают параллельно, система даже больше подвержена неисправностям из-за индуцированного стрессом влияния износа. Поэтому даже тройное резервирование не является эффективным средством защиты от постоянной ошибки, вызванной износом. Для таких целей используется восстановление работоспособности путем замены дефектных модулей на исправные из числа резервных ресурсов.

Методы встроенного восстановления и самовосстановления разработаны и эффективно применяются для регулярных структур, таких как блоки памяти и программируемые вентильные матрицы, а также для регулярных структур аппаратных средств, цифровых фильтров [7-18]. Однако восстановление нерегулярной логики является трудно реализуемым. Целесообразными представляются дополнительные затраты на восстановление работоспособности переключателей и конфигурационной логики, где минимальный размер основного блока в среднем составляет около 200 транзисторов. Обнаружение неисправности, диагностирование и последующий ремонт путем замены дефектного модуля на исправный резервный требует значительных затрат времени. Указанные операции могут быть выполнены во время переключения системы из штатного режима в режим тестирования и восстановления работоспособности. Для обеспечения ремонта в системе должны находиться в «горячем» состоянии избыточные резервные элементы, средства дублирования/тройного резервирования или обработки ошибок на основе специального кода. Это означает, что для обнаружения ошибок онлайн и выполнения ремонтных операций необходимы дополнительные ресурсы. Добавление в проект нескольких резервных модулей и поддержание их в горячем состоянии требует определенных затрат на протяжении всего жизненного цикла системы, в том числе и дополнительной энергии. Решением данной проблемы может быть архитектура, которая обеспечивает контроль и избыточность только тогда, когда это необходимо. В этом случае проверка ошибок и резервная

избыточность могут быть активизированы по специальным сигналам только при необходимости.

Базовая архитектура, используемая для самовосстановления, показана на рис. 1 [7]. Конфигурируемый блок (RB) состоит из  $n$  основных однотипных блоков (BB), которые используются для параллельного выполнения  $n$  функций, и одного резервного блока такого же типа. Резервное устройство может быть использовано как для ремонта, так и для переключения функций между функциональными блоками. Набор из двух переключателей и проходных транзисторов в простейшем случае необходим для каждого входа и выхода. Базовая архитектура не содержит никаких дополнительных элементов, таких как компараторы. Однако если переключательная схема модифицирована таким образом, что два блока могут работать параллельно с одинаковыми входными сигналами, то доступно больше функций до тех пор, пока резервный блок не использован для ремонта. В том случае, если основной блок может работать параллельно с двумя другими блоками, например, BasicBlock2 (BB2) с BB1 либо BB3, возможно также диагностирование неисправностей. При автономном тестировании любой из основных блоков может работать параллельно с одним из других блоков. Тогда при работе, например, BB1 и BB2, BB2 и BB3 или BB3 и запасного блока параллельно с одинаковой функцией, неисправный блок может быть обнаружен при автономном тестировании.

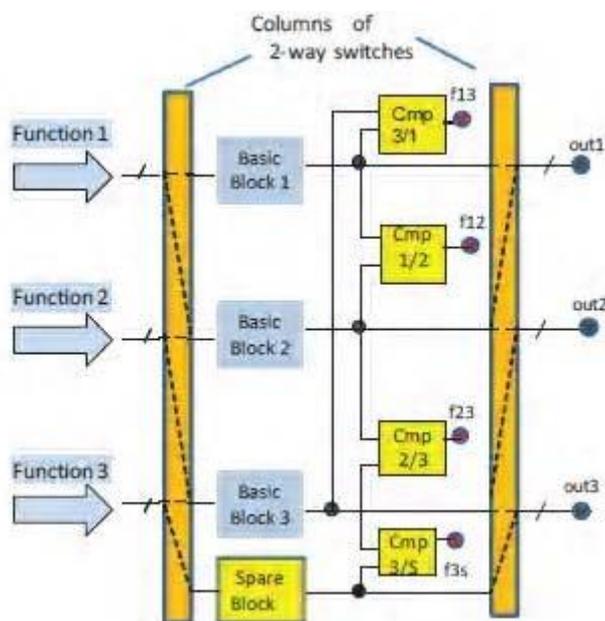


Рис. 1. Реконфигурируемый блок, содержащий один резервный элемент (архитектура 1)

Архитектура со значительно расширенными возможностями показана на рис. 2. Она включает в себя два резервных элемента и трехканальные переключатели на входах и выходах, позволяющие переключать функции прямо, вверх и вниз. Такое переключение и дополнительный резервный блок позволяют запускать две входных функции параллельно на двух основных блоках для проверки онлайн. Кроме того, одна функция может быть выполнена на трех блоках параллельно для проверки онлайн и коррекции ошибок. Два запасных блока могут быть использованы для устранения до двух константных неисправностей, в том числе и производственных дефектов, проявляющихся в процессе эксплуатации.

В обеих описанных выше архитектурах есть два существенных ограничения: отсутствует постоянная онлайн проверка всех сигналов; после обнаружения ошибок процессы диагностирования и ремонта должны быть реализованы в автономном режиме или при запуске, или в течение периода времени, когда система находится «в покое». Усовершенствованная для обеспечения тестирования онлайн и коррекции ошибок архитектура показана на рис. 3.

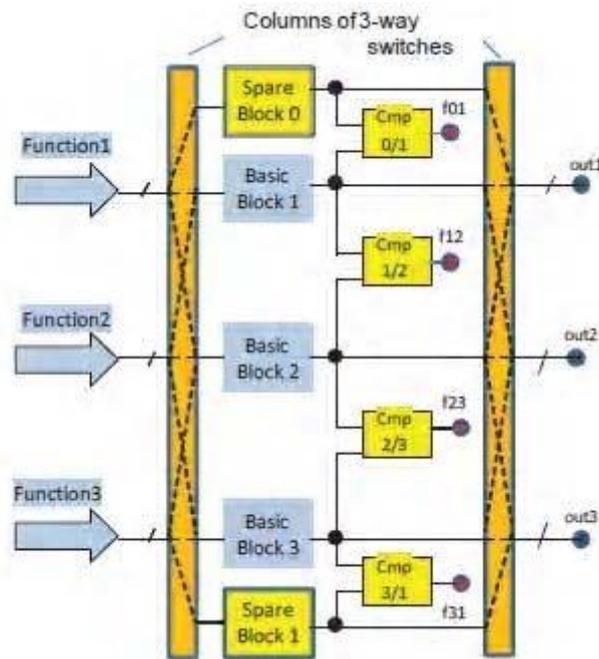


Рис. 2. Реконфигурируемый блок с двумя резервными элементами (архитектура 2)

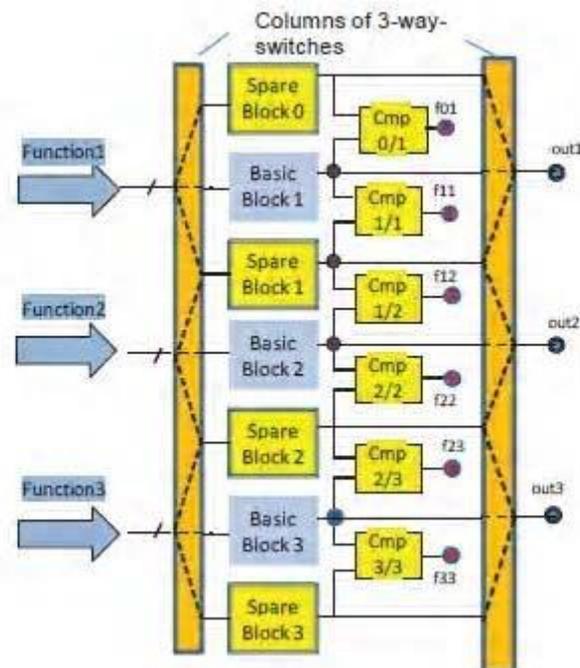


Рис. 3. Реконфигурируемый блок с параллельными резервными элементами (архитектура 3)

Здесь имеется большее количество резервных элементов. Каждый функциональный блок может быть подключен прямо, сверху или снизу. Первый способ функционирования используется только для основных блоков с запасными элементами в режиме ожидания (неактивные) для последующего смещения функций между блоками в целях отключения нагрузки или ремонта. Система с запасными блоками, переведенными в активное состояние для параллельной работы, позволяет выполнять полное обнаружение ошибок онлайн. Вторым режимом работы является обнаружение неисправностей с использованием всех, кроме одного, основных элементов. Один дополнительный запасной блок может использо-

ваться с двойной целью: облегчить параллельную работу после ремонта/замены одного неисправного блока или избирательное тройное резервирование одной функции для выборочной коррекции ошибок.

Наиболее сложная и затратная архитектура показана на рис. 4. Здесь все функции могут работать при тройном резервировании для обнаружения ошибок онлайн. Четыре переключателя для входа/выхода обеспечивают избирательное тройное резервирование (Triple modular redundancy, TMR) даже при наличии одиночной или кратной константной неисправности.

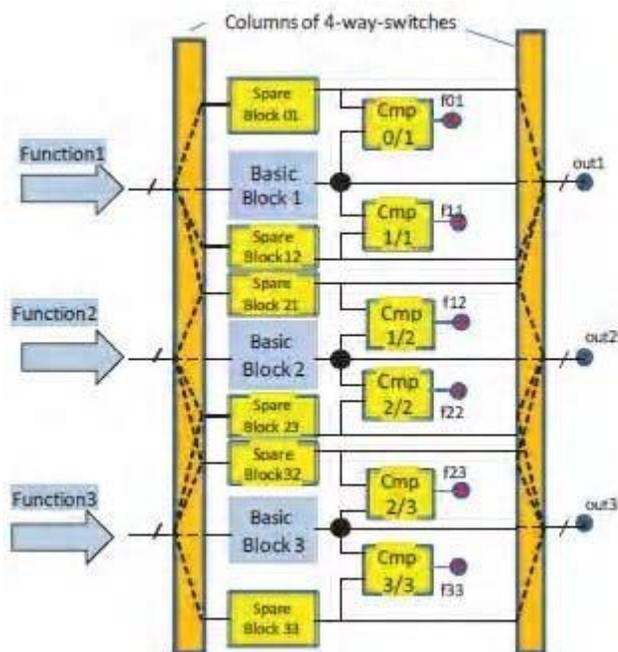


Рис. 4. Реконфигурируемый блок с двойными параллельными резервными элементами (архитектура 4)

Тройное резервирование является наиболее распространенной архитектурой для обнаружения ошибок онлайн и их исправления. Однако для многих приложений связанные с этим накладные расходы непомерно высоки. Избирательное TMR для конкретных сигналов является одним из вариантов решения данной проблемы. Для многих типов интегрированных систем можно ввести классификацию сигналов в зависимости от временных ограничений с одной стороны и уровня безопасности критической функциональности – с другой. В работе выделены три уровня сигналов (рис. 5): 1) нет жестких временных ограничений, безопасность не критична; 2) нет жестких временных ограничений, безопасность критична; 3) есть жесткие временные ограничения, безопасность критична.

Быстрая реакция означает, что необходим отклик в пределах одного такта. Только сигналы уровня 3 требуют постоянного контроля в виде тройного резервирования; для сигналов уровней 1 и 2 TMR не требуется. Например, сигналы уровня 2 могут допускать небольшую задержку с последующей коррекцией ошибки. Такое поведение необходимо для перемежающихся и константных неисправностей, появляющихся во время работы устройства. Только константные неисправности требуют дополнительного процесса самовосстановления с использованием избыточных ресурсов, который должен быть выполнен в автономном режиме.

Архитектуры типов 2 или 3 допускают селективное TMR при возникновении неисправности. В архитектуре 3 обнаружение неисправностей осуществляется путем дублирования и сравнения. Добавление третьего элемента выполняется по требованию после обнаружения неисправности с последующей TMR-операцией. По сути, дополнительное время тратится в обмен на дополнительное оборудование (архитектура 3 против 4). При этом принципиальная схема требует некоторых изменений (дополнительные накладные расходы). В случае появления ошибки она должна быть исправлена и функциональный выход сначала должен

быть изолирован. Для выполнения данной операции может быть использован С-элемент Мюллера (Muller-C-element) (рис. 6).

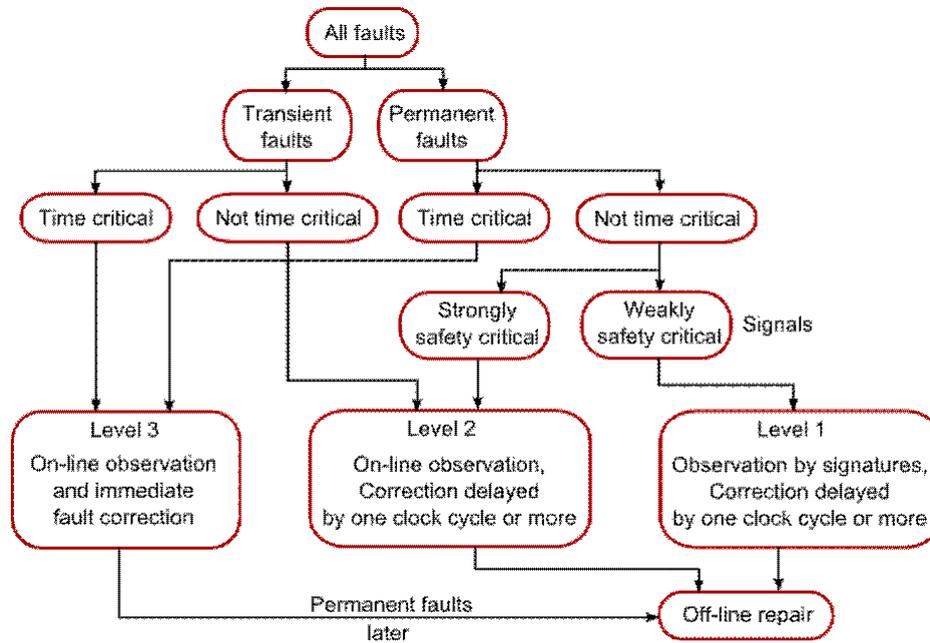


Рис. 5. Классификация сигналов и операций восстановления

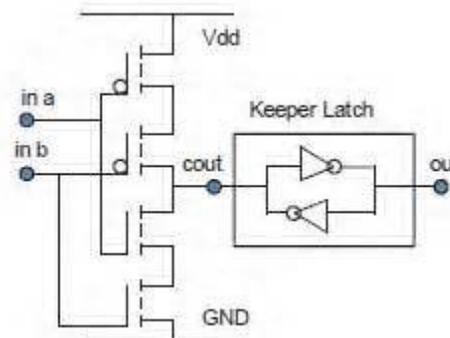


Рис. 6. С-элемент Мюллера

Если входы  $a$  и  $b$  отличаются, выход  $cout$  отделяется от  $Vdd$  и  $GND$  и становится плавающим. Тогда запорная защелка (keeper latch) может быть добавлена для создания определенного логического уровня на выходе. Избирательная схема, основанная на С-элементе Мюллера, показана на рис. 7. Например, имеется неисправность в базовом блоке 1. Три основных блока работают параллельно, и только С-элемент Мюллера, объединяющий выходы SpareBlock0 и SpareBlock1, определяет значение на выходе, в то время как другие находятся в состоянии высокого импеданса. Значение на выходе  $out$  устанавливается правильным выходом. Основные архитектуры 1, 2 и 3 могут быть модифицированы путем добавления С-элементов Мюллера на выходы схем.

Режим работы, основанный на архитектуре 3, описан ниже. При нормальной работе два функциональных блока работают параллельно. Их выходы объединены С-элементом Мюллера и подвергаются сравнению. Если значения на выходах различаются, С-элемент Мюллера отделяет результаты от выходов, и компаратор сигнализирует о неисправности. На выходе схемы может быть восстановлено предыдущее значение с помощью запорной защелки (см. рис. 6). При появлении сигнала неисправности соответствующая функция переключается на TMR за счет соседних функций, которые затем работают как единый функциональный модуль.

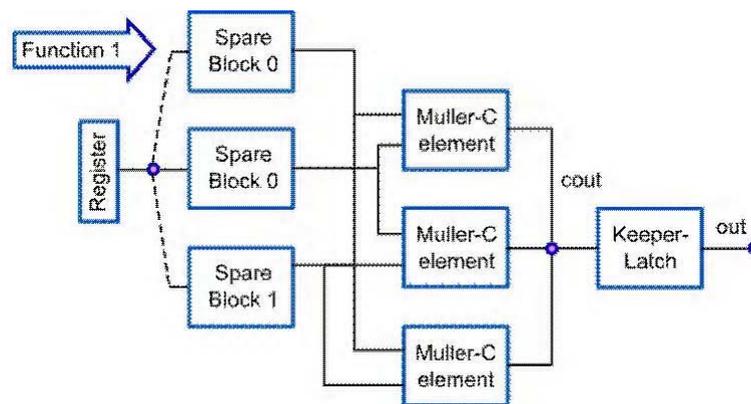


Рис. 7. Выбор с помощью С-элемента Мюллера

Набор из трех С-элементов Мюллера, предназначенный для сравнения выходов двух функциональных блоков, подключается к выходу cout. Только значения на выходах правильно функционирующих функциональных блоков транспортируются на выход схемы (рис. 8).

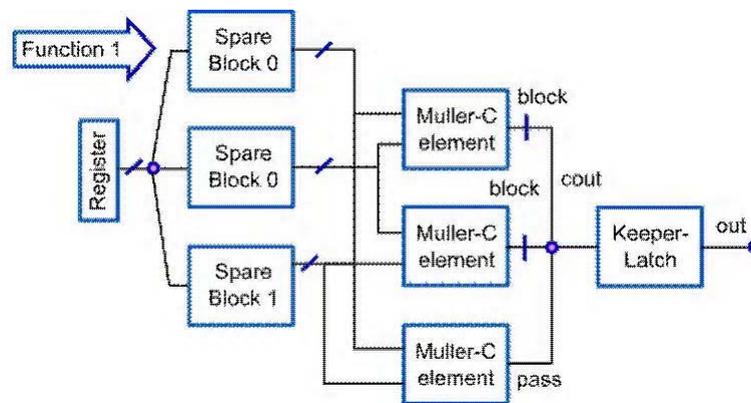


Рис. 8. Избирательное TMR при наличии неисправности

Все операции могут выполняться без прерывания, «правильный» выход дефектной функции появится с некоторой задержкой, которая может сохраняться в течение такта. Расширенная архитектура реконфигурируемого блока, представленного на рис. 2, показана на рис. 9.

В результате исследований [7-9] было определено, что чем больше выбирается размер блоков, на которые разбивается система, тем меньше процент избыточной аппаратуры, необходимой для реализации схемы самотестирования и восстановления. Для 32-битового АЛУ число избыточных элементов равняется 38%, в то время как для вентиля И-НЕ – 200%.

В работах [7-9] упоминается FPGA, но нет привязки к реальной архитектуре программируемой логической матрицы и конфигурируемым блокам CLB, которые являются основными элементами для реализации комбинационной части устройств. Замена одного логического элемента в рамках CLB невозможна, поскольку функция на блоках реализуется в табличном виде. Минимальным блоком для построения самотестируемых схем может быть только CLB.

Описанный метод самотестирования неприменим к отдельным блокам сумматора и устройствам, реализованным на суммировании. Сумматоры строятся на сквозных линиях переноса, идущих вдоль столбцов CLB в FPGA. Исключить из сумматора блок, не нарушив целостности всего устройства, невозможно. Можно выполнять резервирование только для целого компонента. Создание схемы самотестирования на уровне логических элементов приводит к существенной избыточности аппаратуры. Для более высокого уровня сложно

получить однородную схему, которую можно было бы преобразовать в самовосстанавливающуюся систему. В работах [7-9] упоминается метод самовосстановления, основанный на регулярной структуре FPGA, который предлагает исключать из использования столбец или строку CLB в случае обнаружения в нем неисправного блока. Несмотря на то, что метод, как считают авторы, требует улучшения, он является более реальным и технологичным [14]. Кроме того, может быть выполнено резервирование компонентов на уровне столбцов CLB путем внесения дополнительной избыточной теневой конфигурации.

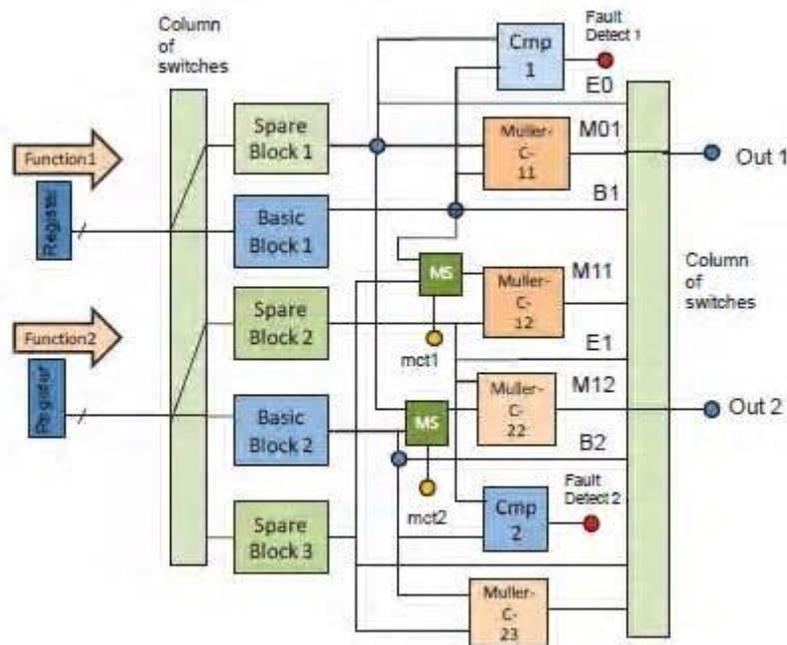


Рис. 9. Реконфигурируемый логический блок с компараторами и С-элементами Мюллера

Встроенный метод восстановления работоспособности мультипроцессорных систем на кристалле предложен в [18]. Ионизирующее излучение может вызывать нежелательные эффекты в полупроводниковых устройствах, например, переключение состояния ячеек памяти (flipping), которые называют мягкими ошибками и относят к распространенному типу нарушений одного события (single event upsets, SEU).

Программируемые логические матрицы FPGA на основе SRAM восприимчивы к неисправностям SEU, при которых нежелательное событие может приводить к однократному кратковременному переключению бита элемента памяти FPGA.

Неисправности SEU в FPGA могут появляться в ячейках встроенной и конфигурационной памяти. Неисправности встроенной памяти повреждают содержимое блока RAM или триггера. Ошибки конфигурационной памяти являются наиболее сложными, поскольку они изменяют поведение FPGA, определяемое набором конфигурируемых логических блоков (Configurable Logic Blocks, CLB) и их межсоединениями.

Межсоединения организованы с помощью переключательных блоков и сегментов трасс, как показано на рис. 10, а. Состояние переключательного блока и сегментов трасс определяется логическим состоянием их конфигурационных битов. Ошибки SEU, влияющие на указанные биты, могут отключить или неправильно подключить сегмент проводника, который соединяет блоки CLB (рис. 10, б).

Упрощенная структура CLB Xilinx FPGA показана на рис. 11. Конфигурируемый блок состоит из набора таблиц преобразования (Look-up Tables – LUTs), триггеров и межсоединений. Источниками неисправностей SEU в CLB являются: 1) биты таблицы преобразований, когда SEU изменяет логическую функцию, имплементированную в look-up table; 2) конфигурационные биты таблицы преобразований; например, биты, устанавливаемые, если ресурс look-up table конфигурирован как таблица преобразований, двойной порт RAM или регистр сдвига; SEU изменяет функциональность таблицы преобразований; 3) мультиплексоры и инверторы, где SEU изменяет внутренние связи CLB.

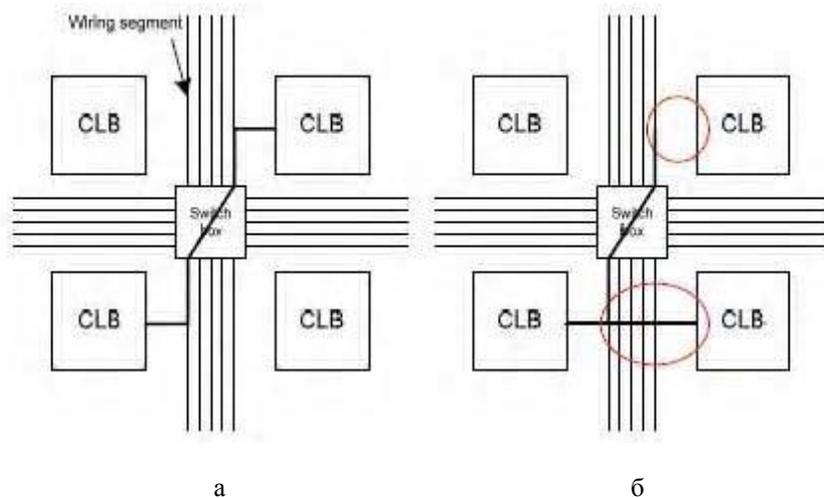


Рис. 10. Межсоединения FPGA

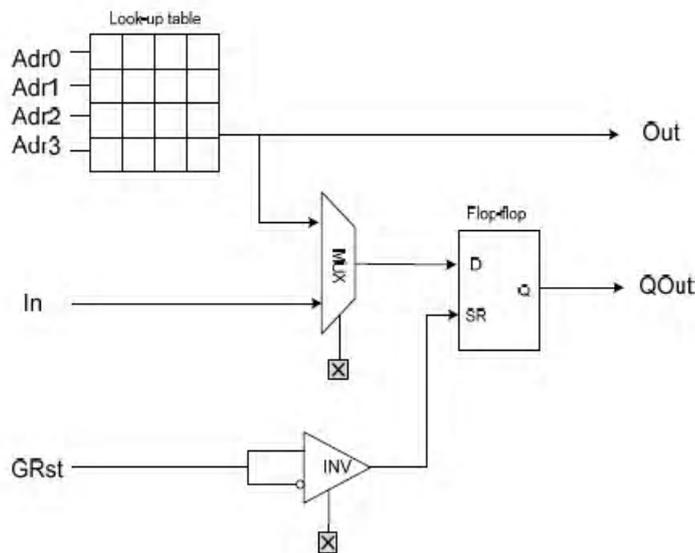


Рис. 11. Источники неисправностей в структуре CLB

Метод коррекции ошибок мультипроцессорных систем на базе FPGA описан ниже. Одно из процессорных ядер сканирует конфигурационные фреймы и выполняет реконфигурацию в случае обнаружения неисправностей (первый проход). Если неисправность появилась непосредственно в процессорном ядре, другое ядро принимает на себя функцию восстановления и выполняет реконфигурацию (расширенное восстановление). Во время нормальной работы системы выбранные процессорные ядра выполняют проверку конфигурации параллельно с выполнением целевого системного приложения. Схемы FPGA являются уязвимыми для SEU и неисправность может возникнуть в любой ячейке конфигурационной памяти в любое время.

Обнаружение и исправление ошибок производится с помощью периферийных ядер с внутренним доступом к конфигурации памяти FPGA. В ПЛИС фирмы Xilinx используется ядро Internal Configuration Access Port (ICAP), которое позволяет встроенному микропроцессору выполнять операции чтения и записи в конфигурационной памяти FPGA с помощью ICAP в реальном времени. Это дает возможность пользователю создавать программное обеспечение для встроенных процессоров в целях изменения структуры и функциональности схемы в процессе ее функционирования. Устройства FPGA Virtex 4 и Virtex 5 включают

в себя средства коррекции ошибок Error Correcting Code (ECC), которые могут быть использованы также для обнаружения ошибок в конфигурационных фреймах. ECC применяют алгоритм кода Хемминга, который определяет местонахождение одиночной неисправности и обнаруживает кратные ошибки. После считывания информации из конфигурационного фрейма ECC формирует значение синдрома (syndrome value). Для одиночной ошибки 11-битное значение синдрома идентифицирует ошибочный бит фрейма.

Возможные сценарии появления ошибок и их исправления:

- ошибка возникает на процессоре, реализующем функциональность, или периферийном модуле; проверяющий процессор корректирует ошибку;
- кратная ошибка возникает на процессоре, реализующем функциональность, или периферийном модуле; неисправность может быть обнаружена, но не может быть исправлена; выполняется реконфигурирование всей системы;
- ошибка останавливает работу проверяющего процессора; после блокировки по времени сторожевой таймер вызывает следующий процессор, который пытается исправить ошибку;
- проверяющий процессор сигнализирует о неудачном выполнении процедуры самотестирования; процессор останавливается; после блокировки по времени сторожевой таймер вызывает следующий процессор, который пытается исправить ошибку;
- ошибка влияет на синхронизацию по времени; сторожевой таймер обнаруживает ошибку, которая не может быть исправлена; выполняется реконфигурирование всей системы;
- ошибка влияет на глобальные сигналы и конфигурационные регистры FPGA; сторожевой таймер обнаруживает ошибку, которая не может быть исправлена; выполняется реконфигурирование всей системы;
- аппаратная ошибка, постоянная неисправность схемы, которая не может быть устранена путем реконфигурации; в случае обнаружения одиночной ошибки запускается алгоритм восстановления работоспособности фрейма, если восстановление не выполнено, то выдается сообщение об аппаратной ошибке.

Аппаратная архитектура приведена на рис. 12. Она включает в себя следующие компоненты: мультиядерная система с  $n$  встроенными микропроцессорными ядрами; внутренний интерфейс частичной реконфигурации, общий доступ к которому имеют все процессорные ядра; механизм взаимного исключения для обеспечения возможности исключаящего доступа к разделяемым периферийным модулям; контроллер прерываний; внешний сторожевой таймер.

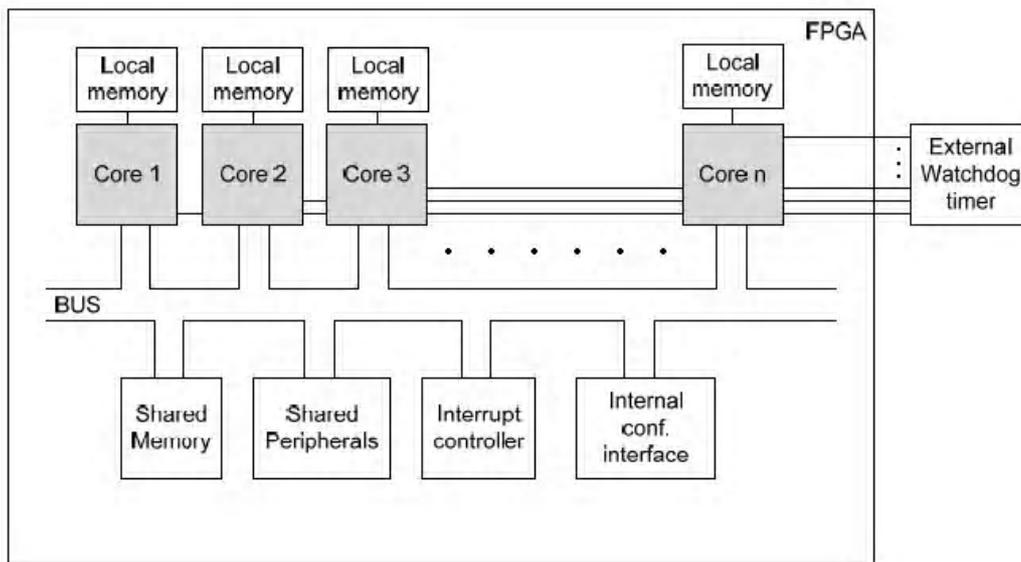


Рис. 12. Архитектура аппаратной платформы

### 3. Заключение

Выполненный обзор технологий восстановления работоспособности мультипроцессорных систем на кристаллах позволяет определить спектр задач дальнейшего совершенствования инфраструктуры встроенного сервисного обслуживания цифровых систем на кристаллах в целях уменьшения времени восстановления работоспособности, что позволяет существенно повысить качество изделия или выход годной продукции. Необходимо разработать метод синтеза функциональных описаний, ориентированных на имплементацию в кристаллы PLD, модели оценивания качества транзакционных соединений в архитектуре вычислительного устройства, автоматную модель переадресации дефектных компонентов комбинационной схемы путем использования ремонтного запаса примитивных элементов, специализированный процессор обработки нечисловых данных, который характеризуется ограниченным набором векторных логических операций, что дает возможность на порядок повысить быстродействие процедур тестирования функциональных нарушений цифровых изделий.

Дальнейшие исследования направлены на повышение качества цифровых изделий за счет встроенного сервисного обслуживания, включающего автономное восстановление работоспособности на основе кубитных моделей функциональностей и специальных технологий диагностирования и ремонта, которые позволяют получить минимальное количество дефектных компонентов цифровой системы на кристалле и минимальное время восстановления работоспособности благодаря нефункциональной программно-аппаратной избыточности, формирующей инфраструктуру SoC.

**Список литературы:** 1. *Dorta T.* Overview of FPGA-Based Multiprocessor Systems / T. Dorta, J. Jimenez, J.L. Martin, U. Bidarte, A. Astarloa // 2009 Intern. Conf. on Reconfigurable Computing and FPGAs. 2009. P. 273 – 278. 2. *Sterpone L.* A New Fault Injection Approach for Testing Network-on-Chips / L. Sterpone, D. Sabena, M.S. Reorda // 20th Euromicro International Conf. on Parallel, Distributed and Network-Based Processing. 2012. P. 530 – 535. 3. *Sharma M.* A novel Test Access Mechanism for failure diagnosis of multiple isolated identical cores / M. Sharma, A. Dutta, Wu-Tung Cheng, B. Benware, M. Kassab // IEEE International Test Conference (ITC). 2011. P. 1 – 9. 4. *Chakrabarty K.* Toward Bug-free Multicore SoC Architectures: System Validation with Transaction-Level Models / K. Chakrabarty // IEEE Design & Test of Computers. – 2011. Vol. 28, Issue 3. P. 4. 5. *Mishra P.* Guest Editors' Introduction: Multicore SoC Validation with Transaction-Level Models / Prabhat Mishra, Sandeep Shukla, Zeljko Zilic // IEEE Design & Test of Computers. 2011. Vol. 28, Issue 3. P. 6-7. 6. *Wagner I.* Reversi: Post-silicon validation system for modern microprocessors / I. Wagner, V. Bertacco // Computer Design. ICCD. Oct. 2008. P.307-314. 7. *Koal, T.; Ulbricht, M.; Vierhaus, H.T.* Combining on-line fault detection and logic self repair / Koal, T.; Ulbricht, M.; Vierhaus, H.T. // IEEE 15th Intern. Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS). 2012. P. 288 – 293. 8. *Koal T.* A Concept for Logic Self Repair / T. Koal, D. Scheit, Vierhaus H.T. // 12th Euromicro Conference on Digital System Design. Architectures, Methods and Tools. 2009. P. 621 – 624. 9. *Koal T.* Basic Architecture for Logic Self Repair / T. Koal, H.T. Vierhaus // 14th IEEE Intern. On-Line Testing Symposium. 2008. P. 177 – 178. 10. *Habermann S.* Built-in Self Repair by Reconfiguration of FPGAs / S. Habermann, R. Kothe, H.T. Vierhaus // 12th IEEE International On-Line Testing Symposium. 2006. 11. *Koal T.* Optimal spare utilization for reliability and mean lifetime improvement of logic built-in self-repair / T. Koal, H.T. Vierhaus // 14th International Symposium Design and Diagnostics of Electronic Circuits & Systems (DDECS). 2011. P. 219 – 224. 12. *Goldstein S.* Reconfigurable computing and electronic nanotechnology / S. Goldstein, M. Budiu, M. Mishra, G. Venkataramani // Proc. of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors. 2003. P. 132 – 142. 13. *Lei Sun.* Runtime Self-Diagnosis and Self-Recovery Infrastructure for Embedded Systems / Lei Sun, Yuki Kinebuchi, Tomohiro Katori, Tatsuo Nakajima // Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems. 2009. P. 284 – 285. 14. *Rab M.T.* Improving Memory Repair by Selective Row Partitioning / M.T. Rab, A.A. Bawa, N.A. Touba // 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. 2009. P. 211 – 219. 15. *Vierhaus H.T.* Embedded Diagnostic Logic Test Exploiting Regularity / H.T. Vierhaus, R. Kothe // 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools. 2008. P. 873 – 879. 16. *Koal T.* A scheme of logic self repair including local interconnects / T. Koal, D. Scheit, H.T. Vierhaus // 12th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS '09). 2009. P. 8 – 11. 17. *Koal T.* On the Feasibility of Built-In Self Repair for Logic Circuits / T. Koal, D. Scheit, M. Schulzel, H.T. Vierhaus // IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT). 2011. P. 316 – 324. 18. *Legat U.* On line self recovery of embedded multiprocessor SOC on FPGA using dynamic partial reconfiguration / Uros Legat, Anton Biasizzo, Franc Novak // Information Technology And Control. 2012. Vol. 41. No 2. P. 116-124. 19. *Zorian Yervant.* Tutorial on

EWDTs 2007. Embedding Infrastructure IP for SOC Yield Improvement / Yervant Zorian // Proceedings of the 39th Design Automation Conference.– New Orleans, LA, USA. 2002. P. 709-712. **20.** Zorian Yervant. Test Strategies for System-in-Package / Yervant Zorian // The Plenary Paper of IEEE East-West Design & Test Symposium (EWDTs'08). Lviv, Ukraine. October 9-12, 2008. **21.** Bhattacharya N. SoftwareHardware Hybrid Systems Verification / N. Bhattacharya // Software Testing, Verification and Validation (ICST). 2011. P.435-438. **22.** Zilic Z. Challenges of Rapidly Emerging Consumer Space Multiprocessors / Z. Zilic, P. Mishra, S.Shukla // Design & Test of Computers. May-June 2011. Vol. 28, No. 3. P.52-53. **23.** Kularatna N. Electronic circuit design from concept to implementation / N. Kularatna. CRC press NW. 2008. 504 p. **24.** Abramovici M. BIST-Based Delay-Fault Testing in FPGAs / Miron Abramovici, Charles E. Stroud // Journal of Electronic Testing: Theory & Applications. 2003. Vol. 19, No. 5. P. 549-558. **25.** Ulbricht M. A new hierarchical built-in self-test with on-chip diagnosis for VLIW processors / Markus Ulbricht, Mario Scholzel, Tobias Koal, Heinrich Theodor Vierhaus // Design and Diagnostics of Electronic Circuits & Systems (DDECS). April 2011. P.143-146. **26.** Elm M. BIST: Scan-based Built-In self-diagnosis / M. Elm, H.-J. Wunderlich // Design, Automation & Test in Europe Conference & Exhibition (DATE). March 2010. P. 1243-1248. **27.** Huang Yu-Jen. A built-in self-test scheme for the post-bond test of TSVs in 3D ICs / Yu-Jen Huang, Jin-Fu Li, Ji-Jan Chen, Ding-Ming Kwai, Yung-Fa Chou, Cheng-Wen Wu // VLSI Test Symposium (VTS). May 2011. P.20-25. **28.** Shianling Wu. Logic BIST Architecture Using Staggered Launch-on-Shift for Testing Designs Containing Asynchronous Clock Domains / Wu Shianling, Laung-Terng Wang, Yu Lizhen, H. Furukawa, Wen Xiaoping, W.-B. Jone, N.A. Toubia, Zhao Feifei, Liu Jinsong, Hao-Jan Chao, Li Fangfang, Jiang Zhigang // Defect and Fault Tolerance in VLSI Systems (DFT). Oct. 2010. P. 358-366. **29.** Tsu-Wei Tseng. A Shared Parallel Built-In Self-Repair Scheme for Random Access Memories in SOCs / Tsu-Wei Tseng, Jin-Fu Li // Test Conference. ITC 2008. Oct. 2008. P.1-9. **30.** Novak O. Handbook of testing electronic systems / O. Novak, E. Gramatova, R.Ubar. Czech University Publishing House. 2005. 402 p. **31.** Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. Харьков: ХНУРЭ. 2009. 484с. **32.** Richter K. A formal approach to MpSoC performance verification / K. Richter, M. Jersak, R. Ernst // Computer. 2003. Vol. 36, Iss. 4. P. 60 – 67. **33.** Cordibella S. A HW/SW co-simulation framework for the verification of multi-CPU systems / S. Cordibella, F. Fummi, G. Perbellini, D. Quaglia // IEEE Intern. High Level Design Validation and Test Workshop. 2008. P. 125 – 131. **34.** Khan G.N. Simulation environment for design and verification of Network-on-Chip and multi-core systems / G.N. Khan, V. Dumitriu // IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems. 2009. P. 1 – 9. **35.** Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А. Проектирование и верификация цифровых систем на кристаллах. Verilog & System Verilog: Харьков: Новое слово, 2010. 528с. **36.** Основы технической диагностики / Под ред. П.П.Пархоменко. М.: Энергия, 1976. 460с. **37.** Пархоменко П.П. Основы технической диагностики (Оптимизация алгоритмов диагностирования, аппаратные средства) / П.П. Пархоменко, Е.С. Согомоян / Под ред. П.П. Пархоменко. М.: Энергия, 1981. 320 с. **38.** Da Silva F. The Core Test Wrapper Handbook. Rationale and Application of IEEE Std. 1500™ / F. Da Silva, T. McLaurin, T. Waayers. Springer.2006.XXIX.276 p. **39.** Marinissen E.J. Guest Editors' Introduction: The Status of IEEE Std 1500 / E.J. Marinissen, Yervant Zorian // IEEE Design & Test of Computers. 2009. No26(1). P.6-7. **40.** Ubar R. Embedded diagnosis in digital systems / R. Ubar, S. Kostin, J. Raik // 26th International Conference "Microelectronics", MIEL 2008. 2008. P. 421-424. **41.** Elm M. Scan Chain Organization for Embedded Diagnosis / M. Elm, H.-J. Wunderlich // Design, Automation and Test in Europe, DATE '08. 2008. P. 468–473. **42.** Автоматизация диагностирования электронных устройств / Ю.В.Малышенко и др. / Под ред. В.П.Чипулиса. М.: Энергоатомиздат, 1986. 216с. **43.** Хаханов В.И. Встроенное диагностирование цифровых систем / В.И. Хаханов, С.В. Чумаченко, Y. Tiecoura, С.С. Галаган // Радіоелектронні і комп'ютерні системи. 2009. №7(41). С. 314-318. **44.** Hahanov V.I. Embedded Testing for SoC Functionality // V.I. Hahanov, S. Pokrova, Y. Tiecoura, A.A. Gorobets // Proc. of the Xth International Conference CADSM. Lviv-Polyana. 2009. P. 29-33. **45.** Bing-Yang Lin. A Memory Failure Pattern Analyzer for memory diagnosis and repair / Lin Bing-Yang, Mincent Lee, Cheng-Wen Wu // IEEE 30th VLSI Test Symposium (VTS). 2012. P. 234 – 239. **46.** Sha'afi Kabiri P. Effective RT-level software-based self-testing of embedded processor cores / P. Sha'afi Kabiri, Z. Navabi, // IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS). 2012. P. 209 – 212. **47.** Tsertov A. Automatic SoC Level Test Path Synthesis Based on Partial Functional Models / A. Tsertov, R. Ubar, A. Jutman, S. Devadze // 20th Asian Test Symposium (ATS). 2011. P. 532 – 538. **48.** Grosso M. A software-based self-test methodology for system peripherals / M. Grosso, W.J. Perez H, D. Ravotto, E. Sanchez, M.S. Reorda, J.V. Medina // 15th IEEE European Test Symposium (ETS). 2010. P. 195 – 200. **49.** Haghbayan M.H. Architecture design and technical methodology for bus testing / M.H. Haghbayan, Z. Navabi // East-West Design & Test Symposium (EWDTs).2010. P. 504 – 509. **50.** Ecker W. TLM+ modeling of embedded HW/SW systems / W. Ecker, V. Esen, R. Schwencker, T. Steininger, M. Velten // Conference & Exhibition Design, Automation & Test in Europe (DATE). 2010. P. 75 – 80.

Поступила в редколлегию 19.06.2012

**Мурад Али Аббас**, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

**Багдади Аммар Авни Аббас (Baghdadi Ammar Awni)**, аспирант кафедры АПВТ ХНУРЭ, Baghdad University. Научные интересы: техническая диагностика цифровых систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

**Хаханов Владимир Иванович**, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

**Литвинова Евгения Ивановна**, д-р. техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: автоматизация диагностирования и встроенный ремонт компонентов цифровых систем в пакете кристаллов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-421. E-mail: kiu@kture.kharkov.ua.

**Дахири Фарид**, студент факультета КИУ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем и сетей, программирование мобильных платформ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

АННОТАЦИИ  
ASU. 2011.  
№158

---

УДК 658.512.011:681.326:519.713

*МУРАД АЛИ АББАС, БАГХДАДИ АММАР АВНИ АББАС, ХАХАНОВ В.И., ЛИТВИНОВА Е.И., ДАХИРИ ФАРИД*

ТЕХНОЛОГИИ ВОССТАНОВЛЕНИЯ РАБОТОСПОСОБНОСТИ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ НА КРИСТАЛЛАХ

Проводится обзор мультипроцессорных систем на кристаллах (MPSoC) и технологий восстановления их работоспособности. Описываются архитектуры MPSoC, базовая архитектура самовосстановления, метод обнаружения и исправления ошибок в программируемых логических матрицах FPGA.

---

ABSTRACTS  
ASU. 2011. №159

---

---

UDC 658.512.011:681.326:519.713

Technologies for repairing multiprocessor systems-on-chips / Murad Ali Abbas, Bahdadi Ammar Avni Abbas, V.I. Hahanov, E.I. Litvinova, Dahiri Farid // Management Information System and Devices. 2012. N 159. P'10-23'

An overview of multiprocessor systems-on-chips (MPSoC) and technologies for their repairing is presented. The MPSoC architectures, the basic architecture of self-repairing, and method for detecting and correcting errors in programmable logic arrays FPGA are described.

Fig. 12. Ref.: 50 items.

---

---

РЕФЕРАТИ  
ASU. 2011. №159

---

---

УДК 658.512.011:681.326:519.713

Технології відновлення працездатності мультипроцесорних систем на кристалах / Мурад Алі Аббас, Багдаді Аммар Авні Аббас, В.І. Хаханов, Є.І. Литвинова, Дахіри Фарід // АСУ та прилади автоматики. 2012. Вип. 159. С. 10-23.

Наведено огляд мультипроцесорних систем на кристалах (MPSoC) і технологій відновлення їх працездатності. Описано архітектури MPSoC, базову архітектуру самовідновлення, метод виявлення та виправлення помилок в програмованих логічних матрицях FPGA.

Л. 12. Бібліогр.: 50 назв.

---

---