

ТЕХНОЛОГИИ ДИАГНОСТИРОВАНИЯ И ВОССТАНОВЛЕНИЯ SYSTEM-IN-PACKAGE

Рассматривается проблема адаптации технологий тестирования цифровых систем на кристаллах (System on Chip – SoC) для нового конструктивного поколения цифровых систем – System-in-Package (SiP), позволяющего эффективно и компактно имплементировать в кристаллы сверхсложные специализированные вычислительные и радиочастотные устройства для рынка электронных технологий. Вместе с тем пакет кристаллов формирует спектр новых задач сервисного обслуживания SiP-функциональностей в реальном масштабе времени, которое существенно отличается от процессов встроенного диагностирования SoC [1-3]. В связи с этим предлагается алгебрологический метод диагностирования и восстановления работоспособности функциональных логических блоков FPGA, основанный на использовании таблиц неисправностей и их анализе в реальном масштабе времени.

1. Введение

Технологическая конструкция SiP, как наслоение или «пирог» из силиконовых кристаллов, закономерно и ожидаемо эволюционирует в течение последних 10 лет на рынке электронных технологий и предоставляет разработчикам аппаратуры новые возможности, связанные с повышением производительности цифровых систем, миниатюризацией объема, снижением веса изделия, уменьшением задержек распространения сигналов, энергопотребления и стоимости устройства.

Указанные преимущества достигаются за счет следующих наиболее существенных и не совсем позитивных свойств. 1) Аппаратное усложнение цифровой системы в пакете, насчитывающей уже миллиарды вентиляей, до 700 миллионов транзисторов на одном кристалле. 2) Высокая стоимость проектирования и производства (Design and Manufacturing Cost). 3) Низкая технологическая привлекательность существующих моделей (Unavailable Simulation Models) и методов анализа и принятия решений (Decision Making Process). 4) Отсутствие гарантии качества (Bare Die Quality) используемых силиконовых пластин. 5) Низкий уровень выхода годных изделий SiP (Low Assembly Yields). 6) Технологическая сложность процессов восстановления работоспособности (Complex Repair Process). 7) Весьма ограниченные средства проектирования (Limited CAD Tools). 8) Наличие проблем в управлении отводом тепла (Thermal Management).

Новое конструктивное исполнение цифровой системы добавляет разработчикам аппаратуры технологические проблемы тестирования цифровых систем в пакетах (SiP Test Challenges), пластин и силиконовых кристаллов (Wafer/Die Test), подложки (Substrate Test) и функциональных модулей (Module Test). Кроме того, существует много проблемных вопросов, связанных с соединением силиконовых кристаллов (silicon die) в цифровую электронную систему. В первую очередь здесь актуальными представляются: 1) механическая защита внутренностей силиконовых кристаллов; 2) удобства и преимущества использования, сборки и тестирования (handling, assembly, test); 3) рассеивание тепла путем его отвода от горячих компонентов (removal of wasted heat).

Одной из наиболее актуальных на рынке электронных технологий проблем в области тестирования, диагностирования и ремонта цифровых систем, оформленных в виде SoC, SiP, является восстановление работоспособности логики пользователя, которая в 2014 году будет составлять 6% от проекта, размещенного на кристалле. На рис.1 представлена тенденция снижения процентного соотношения логической части в сторону наращивания памяти. Тем не менее, проблема оперативного ремонта отказавших логических элементов в реальном масштабе времени остается нерешенной.

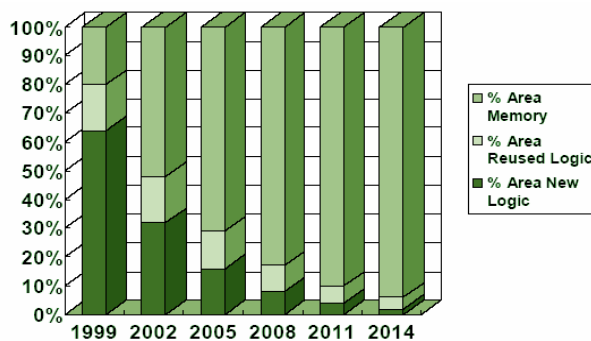


Рис. 1. Тенденция использования памяти и логики в цифровых проектах

Цель исследования – разработка технологии восстановления работоспособности логических компонентов (Reused and New Logic) цифровой системы на кристаллах SoC & SiP в реальном масштабе времени на основе использования избыточных площадей программируемой логики.

Задачи: 1) Анализ конструктивно-технологических особенностей SiP. 2) Анализ методов тестирования, диагностирования и ремонта компонентов SiP. 3) Технологии диагностирования логических блоков цифровых систем в пакетах на основе FPGA (Field Programmable Gate Array). 4) Алгебологический метод восстановления работоспособности логических блоков цифровой системы на основе FPGA. 5) Практические результаты и будущие направления исследования.

За последние годы появились десятки работ, которые рассматривают вопросы, связанные с проблемой тестирования, диагностирования и ремонта цифровых систем на кристаллах и в пакетах (SoC, SiP). Особое место здесь занимает задача тестирования и ремонта логических компонентов цифровых систем ввиду технологической сложности восстановления работоспособности логических элементов при возникновении в них дефектов. Существующие решения, предлагаемые в публикациях, сводятся к следующей классификации:

1. Дублирование логических элементов или областей кристалла, приводящее к удвоению аппаратной реализации функциональности. При фиксации неисправного элемента или области осуществляется переключение на исправный компонент с помощью мультиплексора [4]. Предложенные Xilinx модели FPGA (Field Programmable Gate Array) применимы также при ремонте компонентов FPGA от компании Altera. Основная единица измерения при ремонте – столбец или строка.

2. Использование генетических алгоритмов для диагностирования и восстановления работоспособности на основе автономной реконфигурации кристалла FPGA без использования внешних устройств управления [5]. Надежность диагностирования дефектов равна 99%, время ремонта – 36 миллисекунд вместо 660 секунд, необходимых для стандартного конфигурирования проекта.

3. Восстановление работоспособности кристаллов FPGA, не критичное ко времени, путем замены локальных CLB на избыточные запасные компоненты предложено в [6,7]. Доступный уровень объединения CLB, подлежащий замене, для критически важных приложений составляет порядка тысячи логических элементов.

Технологии восстановления работоспособности логической части цифровых систем, имплементированных в кристаллы программируемой логики, основаны на существовании или внесении избыточности LUT-компонентов FPGA после выполнения процедур Place and Route. Физические дефекты кристалла, возникающие в процессе его изготовления или эксплуатации, проявляются как логические или временные и приводят к неправильному функционированию цифрового изделия. Дефекты привязываются не только к вентилям или LUT-компонентам, но и к конкретному месту на кристалле. Идея ремонта цифровой структуры сводится к исключению дефектного места при повторном выполнении процедур Place and Route после установления диагноза. При этом возможны две технологии ремонта:

1) Запрет дефектной области путем написания управляющих скриптов для достаточно

длительной процедуры Place and Route, что не всегда приемлемо для цифровых систем, работающих в реальном масштабе времени. Однако такой подход приемлем и ориентирован на исключение дефектных областей любой кратности, имеющих место на кристалле. Запрет таких областей при повторном выполнении процедуры Place and Route приводит к восстановлению работоспособности. 2) Для цифровых систем реального времени выполнение процедуры Place and Route, направленной на восстановление работоспособности, может привести к катастрофическим последствиям. Необходим технологический подход, способный восстановить функциональность цифровой системы за миллисекунды, необходимые для перепрограммирования FPGA путем подачи нового битстрима, исключая дефектные области из формирования функциональности. Такой подход основан на предварительной подготовке всех возможных битстримов, изолирующих будущие дефектные области путем их попадания в избыточную нефункциональную область кристалла. Чем больше резервная область, тем меньше число битстримов – проектных вариантов, привязанных к топологии, которое необходимо сгенерировать априори. Что касается кратных дефектов, не покрываемых одной резервной областью, здесь следует сегментировать цифровой проект, разбив его предварительно на непересекающиеся части, которые имеют собственные карты Place and Route. В данном случае можно ремонтировать цифровую систему, которая имеет для n распределенных на пластине дефектов n резервных сегментов. В этом случае общая площадь кристалла представляет собой $n+m$ одинаковых частей.

2. Технологии восстановления работоспособности SiP

Кристаллы FPGA широко используются для быстрого прототипирования в целях уменьшения стоимости проектирования сложных цифровых систем [7]. Модульная структура FPGA позволяет выполнять дистанционное перепрограммирование для замены дефектных логических ресурсов (блоков) исправными резервными элементами. Указанное свойство дает возможность существенно повысить отказоустойчивость аппаратных приложений. Особенно это важно для систем, эксплуатируемых в критических условиях (космическое пространство, экологически опасная для человека среда). Естественно, что промышленно выпускаемые FPGA могут быть подвергнуты полному тестированию еще до реализации этапа их программирования. После имплементации в кристалл заданной функциональности выполняется оперативный контроль FPGA и автономное тестирование компонентов, благодаря наличию специальных встроенных средств.

Систематические (константные) и перемежающиеся (неустойчивые) неисправности могут быть обнаружены и локализованы путем использования встроенных методов тестирования. Использование определенной части ресурсов FPGA в качестве резервных позволяет заменить дефектную область, в которой обнаружена систематическая неисправность. Замена происходит путем перепрограммирования кристалла при сохранении его функциональности. Схема, по которой резервные ресурсы распределяются внутри FPGA и, следовательно, алгоритм реконфигурирования зависят от типа кристалла. Использование процесса частичного конфигурирования позволяет существенно уменьшить среднее время восстановления работоспособности и длину битового потока (bitstream), который используется для реконфигурирования FPGA. Для выбора эффективной стратегии распределения резервных ресурсов большое значение имеет структура межсоединений кристалла.

В опубликованных работах [4, 8-13] описаны различные технологии восстановления работоспособности FPGA, однако сравнительный анализ представленных методов и метрики оценки их эффективности здесь практически отсутствуют. Поэтому далее рассматривается объект исследования, представленный в виде FPGA, его структура и особенности применения, а также существующие методы диагностирования и ремонта, использующие специфику перепрограммирования для устранения дефектов.

Кристалл FPGA есть массив комплексных логических блоков (CLB – Complex Logic Block), каждый из них представляет собой аппаратную реализацию от двух до восьми таблиц истинности на 4 входа, каждая из которых соединена с триггером с помощью ресурсов трассировки, представляющих собой программируемые матрицы переключателей (PSM). В целях обеспечения отказоустойчивости и надежности FPGA выполняется резервирование функциональной части CLB путем априорного формирования подмножества запасных блоков (spares), мощность которого зависит от структуры межсоединений кристалла, влияющих на процедуры ремонта блоков FPGA.

Первый тип межсоединений ориентирован на матричную топологию CLB, когда оптимальная структура функциональности представлена соединением соседних по вертикали и горизонтали логических блоков. Если дефектный и резервный блоки не являются соседними в «манхэттенской» системе координат, то для восстановления работоспособности первого CLB необходимо выполнить достаточно сложную процедуру трассировки, используя матрицы переключателей. Данная структура идеально подходит для обеспечения гибкости при выборе CLB при распределении резервных ресурсов. Недостатки: 1) Большое количество блоков, через которые проходит сигнал, приводит к снижению производительности системы путем увеличения длины логического пути, который зависит от числа используемых промежуточных переключательных матриц. 2) Процесс перепрограммирования межсоединений или трассировки является достаточно сложной задачей, когда замена дефектного логического блока требует значительного количества времени (десятки минут) для поиска квазиоптимального пути.

Второй тип межсоединений представлен на рис. 2. Здесь нижний уровень иерархии связей позволяет соединить соседние блоки CLB. Группы блоков используют межсоединения высшего уровня иерархии, что позволяет уменьшить количество промежуточных переключательных матриц. Данная структура дает возможность эффективно использовать мозаичную (tile) и иерархическую стратегии распределения резервных ресурсов [7].

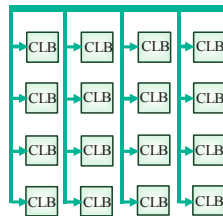


Рис. 2. Иерархическая структура FPGA

Третий тип межсоединений основан на сегментации блоков CLB с помощью программируемых матриц переключателей, разбивающих логические компоненты на подмножества, как показано на рис. 3. Здесь имеются 4 сегмента по 9 логических блоков в каждом. Такая структура не требует выполнения процедуры перетрассировки, если резервный и дефектный блоки расположены на одной линии, вертикальной или горизонтальной, в пределах одного сегмента. В этом случае может быть использован «мозаичный» подход [7], позволяющий использовать резервный CLB для восстановления работоспособности соседнего по горизонтали или вертикали дефектного блока в пределах одного сегмента.

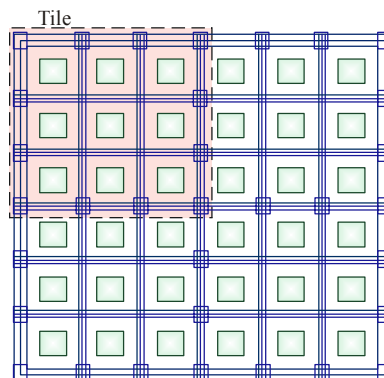


Рис. 3. Частичная сегментация связей в структуре FPGA

Интересным представляется также решение задачи, связанной с выбором стратегии восстановления, которая сводится к созданию протокола перепрограммирования FPGA. Здесь следует рассмотреть два метода частичного реконфигурирования. Первый позволяет определить ресурсы Atmel AT40K FPGA, которые должны быть перепрограммированы, используя пару управляющих регистров для выбора координаты конкретного блока в виде номера строки и столбца реконфигурируемого ресурса. После программирования новая

конфигурация ресурсов может быть загружена в управляющие регистры. Данная стратегия характеризуется незначительным временем исправления ошибки и высокой степенью детализации в процессе реконфигурирования кристалла. Второй метод, используемый компанией Xilinx, заключается в разделении на столбцы перепрограммируемых ресурсов, что может обеспечивать степень детализации (ячеистость – granularity) не хуже, чем в предыдущем случае.

Для устранения постоянной во времени неисправности в кристалле FPGA используется модель процесса восстановления работоспособности, которая представлена на рис. 4.

Шаг 1. Обнаружение неисправностей в FPGA. До начала реконфигурирования выполняется процесс их диагностирования. Тестирование неисправностей происходит с использованием самопроверяемых схем (self-checking circuits). Приложение, имплементируемое в FPGA, разделяется на части, к каждой из них добавляется избыточность, в результате чего функциональный модуль становится самопроверяемой схемой относительно дефектов, которые могут возникнуть внутри модуля. Степень детализации, или глубина диагностирования неисправностей, определяется количеством CLB, подозреваемых в наличии дефектов. Их поиск осуществляется в процессе непрерывного и последовательного циклического тестирования структурных компонентов FPGA, каждый из которых представляет собой совокупность блоков, объединенных в область самотестирования (self-testing areas – STARs). В каждый конкретный момент выполняется проверка только одной области, которая выводится из штатного режима путем ее замены резервом, в то время как оставшаяся часть FPGA продолжает функционировать в нормальном режиме. После завершения процедуры тестирования одной области осуществляется реконфигурирование FPGA в целях функционального отключения следующей области, подлежащей тестированию, и возврата проверенного структурного компонента в штатный режим. Описанная процедура позволяет автоматически устранять перемежающиеся (transient) неисправности с помощью конфигурационной памяти FPGA, которая сохраняет информацию о тестируемой области. Поскольку кристалл постоянно реконфигурируется для тестирования очередного компонента функциональности FPGA, процедура диагностирования имеет высокую степень детализации или глубину поиска дефектов, определяемую не более чем 10 CLB. Недостатком метода является зависимость времени диагностирования дефекта от длительности цикла тестирования функциональных компонентов FPGA, которая задается множеством самопроверяемых областей FPGA, полученным на стадии тестопригодного проектирования. Кроме того, наличие задержки между двумя событиями – возникновением и обнаружением дефекта – может приводить к неверному диагнозу о работоспособности системы в определенный момент времени, что приводит к нежелательным последствиям для критических систем реального времени.

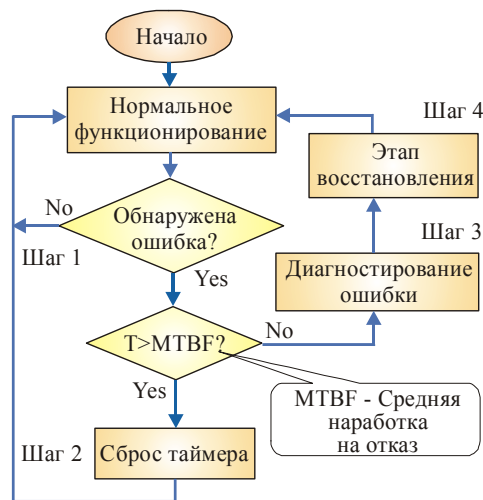


Рис. 4. Процедура восстановления работоспособности FPGA

Шаг 2. Позволяет различить перемежающиеся и систематические (константные) неисправности. Когда устройство диагностирует факт наличия дефекта, выполняется обновление конфигурационной памяти FPGA. Затем выполняется инициализация таймера, контролирующего MTBF (Mean Time Between Failures – средняя наработка на отказ), для различения перемежающихся и систематических неисправностей. Если две ошибки обнаружены в одном месте в течение интервала времени меньшего, чем MTBF, фиксируется факт наличия систематической неисправности с последующим выполнением шагов 3 и 4.

Шаг 3. В случае обнаружения систематической неисправности выполняется ее диагностирование со степенью детализации лучшей, чем при разделении схемы на самотестируемые модули. Для поиска дефектного CLB может быть использован метод, предложенный в [14].

Шаг 4. Ремонт дефектного модуля CLB. Механизм восстановления работоспособности непосредственно зависит от архитектуры FPGA. При этом используются различные технологии в зависимости от возможностей частичной или динамической реконфигурации FPGA, а также от структуры битового потока для перепрограммирования кристалла и межсоединений: 1) Иерархическая модель предполагает формирование двух уровней избыточности. На нижнем уровне выполняется разделение FPGA на клетки (tile) – подматрицы блоков CLB, в каждой из них создаются резервные блоки. На верхнем уровне дефектные клетки могут быть заменены резервными клетками, как показано на рис. 5,а. 2) Оптимальная модель предполагает, что резервные CLB могут быть использованы для восстановления работоспособности любого дефектного модуля цифровой системы на кристалле FPGA. При этом нет необходимости выполнять времязатратные процедуры перетрассировки (рис. 5, б). Данный подход инвариантен к структуре FPGA, но требует существенного времени восстановления работоспособности и дополнительного объема памяти для хранения предварительно скомпилированного битового потока (bitstream), перепрограммирующего кристалл.

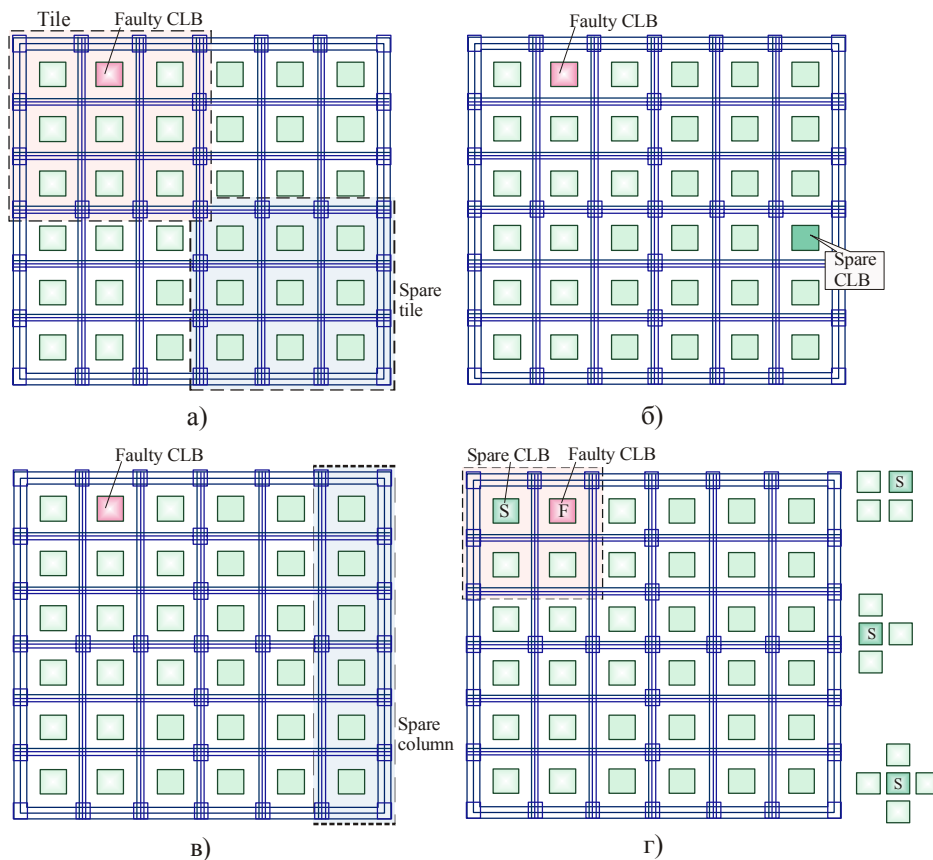


Рис. 5. Модели FPGA для восстановления работоспособности

Использование bitstream является обязательным, поскольку процедура перепрограммирования может затрагивать все ресурсы FPGA. Следовательно, должен быть реализован полный алгоритм place-and-route. Данная процедура является времязатратной и не может быть выполнена в режиме on-line. Она реализуется во время компиляции проекта с использованием методов уменьшения размерности предварительно скомпилированного битового потока. 3) Модель с избыточностью оперирует матрицей блоков CLB, разделенной на столбцы, среди которых имеется один или несколько избыточных, используемых для восстановления работоспособности цифровой системы (рис. 5, в). Если неисправность обнаружена в функциональном столбце, он маркируется как дефектный и заменяется резервным. Модель процесса позволяет выполнять частичное реконfigurирование структуры FPGA путем разделения bitstream, как это используется в кристаллах компании Xilinx. Процедура реконfigurирования выполняется в реальном масштабе времени. Также имеется возможность упростить, а значит – ускорить ее реализацию с помощью укрупнения детализации функциональных модулей. Недостаток: при обнаружении дефектного CLB другие исправные блоки, принадлежащие столбцу, также маркируются как дефектные и выводятся из функционирования. 4) Мозаичная модель. Здесь структура FPGA делится на клетки, содержащие резервные CLB, каждый из которых может восстановить один дефектный блок в рассматриваемой клетке (рис. 5, г).

В процессе диагностирования определяется местоположение дефектной области с глубиной детализации выше, чем размер клетки (ячейки). Это дает возможность заменить дефектный блок CLB резервным элементом клетки. Реконfigurация клетки обеспечивает исходную функциональность на новой карте. Межсоединения по периметру клетки с остальной частью FPGA должны быть неизменными в процессе реконfigurации. Такой подход позволяет уменьшить время восстановления работоспособности цифровой системы после обнаружения дефекта. Запасные конфигурации генерируются на стадии проектирования и хранятся в памяти. Каждая клетка формируется как набор блоков CLB и межсоединений FPGA с помощью описания интерфейса, который определяет и обеспечивает связи с другими клетками кристалла. Применение интерфейса клеток позволяет не вовлекать другие компоненты в процесс реконfigurирования, уменьшая используемую память. Модель позволяет восстанавливать не только CLB, но и локальные дефектные межсоединения. Ошибки глобальных межсоединений требуют других подходов, поскольку они пересекают периметр клеток, делая их зависимыми друг от друга. Структура клетки зависит от межсоединений FPGA. Как правило, один из нескольких CLB, принадлежащих клетке, является резервным элементом. Остальные блоки используются в штатном режиме. При обнаружении неисправности клетка реконfigurруется путем исключения из функционирования дефектного блока. На рис. 5,г показаны структуры клетки, форма которых отличается от квадрата. В кристаллах Atmel FPGA используются диагональные межсоединения блоков, для которых модель, представленная на рис. 5,г, является более предпочтительной [15].

Иерархическая модель представляет собой наиболее общий случай восстановления работоспособности. Все остальные, описанные выше, могут рассматриваться как ее частные случаи. Оптимальная структура применима только на нижнем уровне иерархии, в то время как модель с избыточностью имеет резервные ресурсы только на верхнем уровне для замены дефектного столбца блоков. Мозаичная модель имеет избыточность на нижнем уровне иерархии, поэтому один из двух дефектных блоков, расположенных в одной клетке, является невозстанавливаемым. Для устранения данного недостатка следует использовать дополнительные резервные клетки.

При использовании двухуровневой архитектуры цифровой системы в кристалле FPGA вероятность ее исправного функционирования при наличии не более чем q дефектов, равного количеству резервных элементов, определяется выражением [7]:

$$P_{nf}(t) = \sum_{i=0}^q \binom{m}{i} P_{tile}(t)^{m-i} (1 - P_{tile}(t))^i,$$

где m – общее число клеток в FPGA или блоков CLB в клетке; q – количество резервных клеток или блоков в каждой из них.

Для сравнения четырех моделей восстановления работоспособности SiP в [7] использовались матричная (квадратная) структура FPGA (число строк и столбцов равно 100) с резервом CLB, равным 25%. 1) Оптимальная модель: количество CLB, используемых в процессе функционирования системы, равно 7500, резервных элементов – 2500. 2) Мозаичное восстановление: каждая клетка (tile) включает 4 CLB, 3 из которых используются в процессе функционирования и один является резервным, количество клеток равно 2500. 3) Модель с избыточностью: количество столбцов равно 100, каждый столбец состоит из 100 CLB, 75 из которых используется в работе и 25 – резервные. 4) Иерархическая модель: количество клеток равно 2000, каждая из них содержит 5 CLB, 4 из которых – функциональные и один – резервный. Используется 1875 рабочих клеток и 125 резервных. Общее число резервных блоков CLB равно 2500.

Для малых значений интенсивности отказов надежность восстановления с избыточностью превосходит надежность ремонта на основе мозаичной модели. При увеличении интенсивности отказов надежность первого подхода резко падает ввиду ограниченного количества резервных элементов, а надежность второго – плавно уменьшается благодаря меньшей избыточности блоков CLB для восстановления работоспособности. Таким образом, существующие структуры восстановления работоспособности можно разделить на две группы: 1) Модели, ориентированные на замену единичного элемента (оптимальная и мозаичная). 2) Модели, ориентированные на замену группы дефектных элементов (иерархическая и модель с избыточностью).

Реализация рассмотренных в работе моделей восстановления работоспособности цифровых систем основана на выполнении времязатратной процедуры Place and Route или использовании управляющих регистров для оперативного реконфигурирования структуры FPGA в реальном масштабе времени в целях исключения из процесса функционирования дефектных блоков.

3. Технологии диагностирования блоков FPGA

Основная роль в процессе диагностирования неисправных блоков FPGA отводится встроенной в кристалл инфраструктуре граничного сканирования [18], которая ориентирована на решение практически всех задач сервисного обслуживания SiP-функциональностей. Контроллер доступа обеспечивает мониторинг всех проблемных внутренних линий проекта с помощью регистра граничного сканирования, число разрядов которого должно соответствовать наперед заданной глубине диагностирования, а следовательно – количеству блоков LUT, клеток или кристаллов в SiP. Модель процесса сервисного обслуживания SiP-функциональности для диагностирования дефектов в компонентах кристаллов представлена на рис. 6.

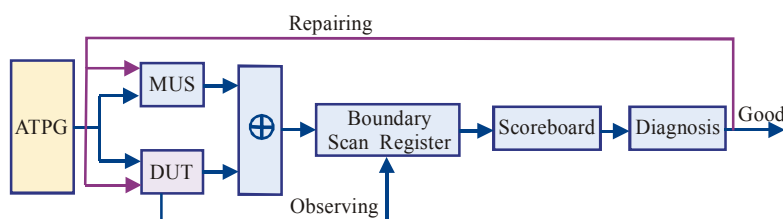


Рис. 6. Модель процесса диагностирования F-IP

Здесь модуль функции Хог анализирует выходные реакции модели MUS и реального устройства DUT на тестовые векторы, поступающие от встроенного генератора входных последовательностей. Блок Boundary Scan Register является мультизондом, предназначенным для мониторинга состояния всех блоков или клеток, далее компонентов кристалла. Модуль Scoreboard выполняет функцию анализа результатов мониторинга для диагностирования и последующего ремонта компонентов SoC. Результат диагностирования SiP представляет собой множество неисправных блоков, подлежащих исключению из функционального режима путем их замены на резервные компоненты.

Интересное решение задачи диагностирования может быть получено путем применения булевой алгебры для анализа таблицы неисправностей (ТН) М, представляющей собой декартово произведение теста Т на множество наперед заданных дефектов F. Вектор

экспериментальной проверки V , равный по длине различным тестовым сегментам, позволяет привести процедуру поиска дефектов к задаче покрытия, что дает максимально точный результат в виде ДНФ, где каждый терм есть возможный вариант наличия в устройстве подмножества или сочетания дефектов SiP. Модель процесса диагностирования представлена в следующем виде:

$$\begin{aligned}
 A &= \langle T, F, M, V \rangle, \\
 T &= (T_1, T_2, \dots, T_i, \dots, T_n); F = (F_1, F_2, \dots, F_j, \dots, F_m); \\
 M &= |M_{ij}|, i = \overline{1, n}; j = \overline{1, m}; V = (V_1, V_2, \dots, V_i, \dots, V_n); \\
 V_i &= R(T_i) \oplus R^*(T_i); \{V_i, T_i, M_{ij}, F_j\} \in \{0, 1\}.
 \end{aligned} \tag{1}$$

Значение координаты вектора V есть результат выполнения операции Хог над обобщенной модельной и фактической реакцией выходов устройства. Если хотя бы на одном выходе устройства зафиксировано единичное значение вектора V на i -м тестовом сегменте, то обобщенное состояние выхода равно 1. В противном случае значение координаты вектора V равно 0.

3.1. Алгебро-логический метод диагностирования неисправностей

Решение задачи диагностирования сводится к анализу таблицы неисправностей, полученной в результате моделирования дефектов компонентов SiP, путем записи логического произведения дизъюнкций строк таблицы неисправностей, записанных по единичным значениям вектора экспериментальной проверки:

$$F = \bigwedge_{\forall V_i=1}^{i=\overline{1, n}} \left(\bigvee_{\forall M_{ij}=1}^{j=\overline{1, m}} F_j \right). \tag{2}$$

Конъюнктивная нормальная форма (КНФ), полученная из ТН, трансформируется к дизъюнктивной нормальной форме (ДНФ) с помощью эквивалентных преобразований (логическое умножение, минимизация и поглощение) [17]. В результате получается булева функция, где термы – логические произведения представляют полное множество решений в виде сочетания дефектов (дающих по выходам SiP или ее компонентам двоичные координаты вектора экспериментальной проверки V):

$$F = \bigwedge_{\forall V_i=1}^{i=\overline{1, n}} \left(\bigvee_{\forall M_{ij}=1}^{j=\overline{1, m}} F_j \right) = \left[\begin{array}{l} a \vee ab = b \\ a \vee a = a \end{array} \right] = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^m k_j F_j \right), k_j = \{0, 1\}. \tag{3}$$

Функция (3) в общем случае формирует диагноз в виде некоторого подмножества сочетаний (кратных) дефектов, которые далее нуждаются в уточнении путем применения дополнительного зондирования внутренних точек с помощью регистра граничного сканирования. Количество единиц в векторе экспериментальной проверки V формирует число дизъюнктивных термов КНФ (3). Каждый терм – построчная запись дефектов (через логическую операцию ИЛИ), оказывающих влияние на выходы функциональности. Представление таблицы в виде аналитической записи – конъюнктивной нормальной формы дает потенциальную возможность существенно сократить объем диагностической информации для поиска дефектов. Последующее преобразование КНФ к ДНФ на основе тождеств алгебры логики позволяет существенно уменьшить булеву функцию.

Алгебро-логический метод рассматривается на примере таблицы неисправностей M_1 и представлен в виде следующих пунктов алгоритма [21].

$$M_1 = \begin{array}{c|cccccc|c}
 T_i \backslash F_j & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & V \\
 \hline
 T_1 & 1 & & & 1 & & & 1 \\
 T_2 & & 1 & & & 1 & & 1 \\
 T_3 & & & 1 & 1 & & 1 & 1 \\
 T_4 & 1 & & 1 & & & & 1 \\
 T_5 & & 1 & & & 1 & 1 & 0
 \end{array}$$

1. Определение строк таблицы неисправностей, соответствующих нулевым значениям вектора экспериментальной проверки в целях обнуления единичных координат найденных строк. В данном случае – это одна строка T5.

2. Нахождение всех столбцов, которые имеют нулевые значения координат строк с нулевым состоянием ВЭП. Обнуление единичных значений найденных столбцов. В данном случае: F2, F5, F6.

3. Удаление из таблицы неисправностей строк и столбцов, имеющих только нулевые значения координат (найденные в пунктах 1 и 2).

$$M_1 = \begin{array}{c|cccccc|c} T_i / F_j & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & V \\ \hline T_1 & 1 & & & 1 & & & 1 \\ T_2 & & 0 & & & 0 & & 1 \\ T_3 & & & 1 & 1 & & 0 & 1 \\ T_4 & 1 & & 1 & & & & 1 \\ T_5 & & 0 & & & 0 & 0 & 0 \end{array} = \begin{array}{c|ccc|c} T_i / F_j & F_1 & F_3 & F_4 & V \\ \hline T_1 & 1 & & 1 & 1 \\ T_3 & & 1 & 1 & 1 \\ T_4 & 1 & 1 & & 1 \end{array}$$

4. Построение КНФ по единичным значениям ВЭП. Преобразование КНФ к ДНФ с последующей минимизацией функции:

$$\begin{aligned} F &= (F_1 \vee F_4) \wedge (F_3 \vee F_4) \wedge (F_1 \vee F_3) = \\ &= (F_1 F_3 \vee F_3 F_4 \vee F_1 F_4 \vee F_4 F_4) \wedge (F_1 \vee F_3) = \\ &= F_1 F_1 F_3 \vee F_1 F_3 F_4 \vee F_1 F_1 F_4 \vee F_1 F_4 F_4 \vee F_1 F_3 F_3 \vee \\ &\vee F_3 F_3 F_4 \vee F_1 F_3 F_4 \vee F_3 F_4 F_4 = \\ &= F_1 F_3 \vee F_1 F_3 F_4 \vee F_1 F_4 \vee F_3 F_4 \vee F_1 F_3 F_4 \vee F_3 F_4 = \\ &= F_1 F_3 \vee F_1 F_4 \vee F_3 F_4. \end{aligned}$$

Предложенный алгоритм ориентирован на анализ таблицы неисправностей в целях уменьшения объема ТН и последующих вычислений, связанных с построением ДНФ, которая формирует все решения по установлению диагноза функциональностей SoC. Дальнейшее уточнение диагноза возможно только с применением мультисонда на основе регистра граничного сканирования данных [18].

3.2. Векторно-логический метод диагностирования неисправностей

Обработка ТН для получения диагноза выполняется по сценарию, основанному на использовании векторных операций конъюнкции, дизъюнкции и отрицания над стоками таблицы неисправностей. Аналитическая векторно-логическая модель диагностирования кратных дефектов определяется конъюнкцией двух компонентов, где первый есть дизъюнкция векторов, соответствующих единичным координатам ВЭП, а второй – инверсия дизъюнкций векторов, соответствующих нулевым координатам ВЭП:

$$F = M^1 \wedge \overline{M}^0 = \left(\bigvee_{v_i=1} M_i \right) \wedge \left(\overline{\bigvee_{v_i=0} M_i} \right). \quad (4)$$

Модель диагностирования одиночных дефектов отличается выполнением на первом шаге сценария операции конъюнкции (вместо дизъюнкции) всех векторов, соответствующих единичным координатам ВЭП:

$$F = M^1 \wedge \overline{M}^0 = \left(\bigwedge_{v_i=1} M_i \right) \wedge \left(\overline{\bigvee_{v_i=0} M_i} \right). \quad (5)$$

Пример 1. Выполнить диагностирование кратных дефектов в структуре SiP векторно-логическим методом, для которой заданы таблица неисправностей и вектор экспериментальной проверки:

T_i / F_j	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	V
T_1				1						1	1
T_2		1					1				0
T_3			1			1			1		0
T_4	1										0
T_5					1			1			1
T_6	1	1									0
T_7			1								0
T_8				1							1
T_9					1	1					0
T_{10}							1				0
T_{11}								1	1	1	1
M^1				1	1			1	1	1	1
M^0	1	1	1		1	1	1		1		0
\overline{M}^0	0	0	0	1	0	0	0	1	0	1	1
F	0	0	0	1	0	0	0	1	0	1	1

(6)

Обработка таблицы неисправностей в соответствии с формулой (4) дает результат, представленный в четырех нижних строках (6). Последняя строка таблицы неисправностей фиксирует факт наличия в схеме дефектов, представленных в векторной или теоретико-множественной форме $F = (0001000101) = \{F_4, F_8, F_{10}\}$.

Для преобразования полученного решения к ДНФ кратных дефектов используется структура ТН, маскируемая вектором F , представленным последней строкой таблицы:

$$F = M^1 \wedge F = \bigwedge_{V_i=1} (M_i \wedge F). \quad (7)$$

Полученные векторы, число которых в пределе равно количеству единиц ВЭП, логически перемножаются. При этом каждый вектор может быть компактно записан в виде дизъюнкции значений единичных координат. Для представленной выше ТН применение формулы (7) дает следующий результат:

$$\begin{aligned} F &= (F_4 \vee F_{10})(F_8)(F_4)(F_4 \vee F_{10}) = \\ &= F_4 F_8 F_4 F_4 \vee F_{10} F_8 F_4 F_4 \vee F_4 F_8 F_4 F_{10} \vee F_{10} F_8 F_4 F_{10} = \\ &= F_4 F_8 \vee F_{10} F_8 F_4 \vee F_4 F_8 F_{10} \vee F_{10} F_8 F_4 = F_4 F_8. \end{aligned}$$

Он интересен тем, что благодаря записи дефектов в виде термов ДНФ, покрывающих все единичные координаты ВЭП, появилась возможность исключить дефект $F_{10} \in F$ из списка неисправностей.

Преимущество векторно-логического метода заключается в технологичности анализа таблицы неисправностей, вычислительная сложность которого имеет мультипликативную зависимость от числа дефектов и мощности теста: $Q = n \times m$. Метод следует использовать при преобладании в таблице неисправностей единичных координат, когда алгебро-логический метод дает высокую оценку по Квайну сложности ДНФ и КНФ. Недостаток заключается в невозможности представления всех сочетаний дефектов, формирующих термы для покрытия единичных координат вектора экспериментальной проверки.

Предложенные методы диагностирования: алгебро-логический и векторно-логический предлагают специалисту в области проектирования и тестирования цифровых систем на кристаллах математический аппарат, который способен осуществить диагноз дефектных компонентов, если имеется предварительно построенная таблица неисправностей. При этом векторные решения, эффективно полученные во втором методе, можно представлять всеми возможными сочетаниями дефектов в виде термов ДНФ, что характерно для первого метода. Второй метод эффективен при числе единиц в матрице неисправностей, превышающем 10-20%.

4. Алгебро-логический метод ремонта блоков FPGA

Представлен точный метод восстановления работоспособности логических блоков FPGA с помощью резервных компонентов путем квазиоптимального решения задачи покрытия множества дефектных ячеек минимально возможным количеством избыточных компонентов. Метод ориентирован на имплементацию в инфраструктуру сервисного обслуживания функциональностей цифровой системы в пакете. Функция цели определяется как минимизация числа резервных компонентов S^t матрицы блоков FPGA, необходимых для ремонта дефектных блоков в процессе функционирования SiP путем синтеза дизъюнктивной нормальной формы покрытия неисправностей с последующим выбором минимального конъюнктивного терма $(S_1^t, S_2^t, \dots, S_i^t, \dots, S_{m_t}^t) \in S^t$, удовлетворяющего ограничениям по числу резервных компонентов $m_t \leq p$, которые входят в состав логического произведения:

$$Z = \min_{t=1, n} (|S^t|) \Big|_{m_t \leq p}, S = (S^1 \vee S^2 \vee \dots \vee S^t \vee \dots \vee S^n); S^* = \{S_1, S_2, \dots, S_j, \dots, S_p\}.$$

Модель функциональности FPGA представляется в виде матрицы логических блоков, оперирующей строками и столбцами структуры $M = |M_{ij}|$. В процессе проектирования матрице придается некоторый резерв, составленный из запасных строк и столбцов, которые можно переадресовывать в процессе реконfigurирования структуры устройства в случае обнаружения дефектов.

Модель процесса определения минимального числа резервных компонентов (строк и столбцов), покрывающих все обнаруженные дефекты в матрице логических блоков FPGA, сводится к следующим пунктам:

1. Построение таблицы покрытия обнаруженных дефектных блоков FPGA резервными строками и столбцами матрицы. Для достижения поставленной цели рассматривается топологическая модель результатов тестирования функциональности FPGA в виде матрицы, координаты которой идентифицируют обнаруженные дефекты (исправные и неисправные блоки):

$$M = |M_{ij}|, M_{ij} = \begin{cases} 1 \leftarrow T \oplus f = 1; \\ 0 \leftarrow T \oplus f = 0. \end{cases} \quad (8)$$

Здесь координата матрицы равна 1, если сумма по модулю 2 функции исправного поведения f блока и фактической тестовой реакции дают единичное значение, что соответствует наличию в блоке дефекта. После тестирования FPGA – фиксации всех неисправных блоков выполняется построение таблицы покрытия дефектов $Y = |Y_{ij}|, i = \overline{1, n}; j = \overline{1, m}$, где столбцы соответствуют множеству неисправностей (единичных координат), зафиксированных в матрице M ($|M| = m$), а строки есть номера столбцов и строк матрицы блоков FPGA, покрывающих неисправности, указанные в столбцах:

$$Y = |Y_{ij}|, Y_{ij} = \begin{cases} 1 \leftarrow C_i(R_j) \cap F_j \neq \emptyset; \\ 0 \leftarrow C_i(R_j) \cap F_j = \emptyset. \end{cases} \quad (9)$$

Оперируя далее строками и столбцами, в которых содержатся дефекты, можно найти оптимальное решение в виде покрытия обнаруженных дефектов в метрике неисправных строк и столбцов. Далее выполняется тривиальная процедура переназначения дефектным строкам и столбцам аналогичных исправных компонентов из резерва кристалла FPGA.

Из соображения технологичности решения задачи покрытия вместо компонентов двумерной метрики, обозначенной строками и столбцами (C, R) , используется одномерный вектор, сконкатенированный из двух последовательностей C и R , мощность которого равна $n = p + q$:

$$\begin{aligned} X &= C * R = (C_1, C_2, \dots, C_i, \dots, C_p) * (R_1, R_2, \dots, R_j, \dots, R_q) = \\ &= X^c * X^r = (X_1, X_2, \dots, X_i, \dots, X_p, X_{p+1}, X_{p+2}, \dots, X_{p+j}, \dots, X_{p+q}). \end{aligned} \quad (10)$$

При этом между элементами исходных наборов (С, R) и результирующим вектором X существует взаимно-однозначное соответствие, установленное в первом столбце матрицы Y. Следует заметить, что преобразование $X = C * R$ выполняется лишь для удобства рассмотрения и последующего построения дизъюнктивной нормальной формы в рамках единообразия переменных, формирующих булеву функцию. Если данную процедуру не выполнять, то функция будет определена на двух типах переменных, содержащих столбцы и строки матрицы памяти.

2. Построение КНФ путем аналитического, полного и точного решения задачи покрытия. После формирования матрицы покрытия, содержащей нулевые и единичные координаты, выполняется синтез аналитической формы покрытия путем записи КНФ по единичным координатам столбцов матрицы. Здесь число конъюнктивных термов равно количеству столбцов таблицы, а размерность дизъюнктивного терма равна числу единиц в рассматриваемом столбце:

$$Y = \bigwedge_{j=1}^m (Y_{pj} \vee Y_{qj})_{\{Y_{pj}, Y_{qj}\}=1} = \bigwedge_{j=1}^m (X_{pj} \vee X_{qj}). \quad (11)$$

Из последнего выражения видно, что каждый столбец идентифицирует два варианта покрытия неисправности – строкой и столбцом. Поэтому столбец имеет только две координаты, имеющие единичное значение, а число логических произведений равно общему количеству дефектов m, обнаруженных в матрице блоков FPGA.

3. Преобразование КНФ к ДНФ дает возможность увидеть все решения задачи покрытия. Для этого к конъюнктивной нормальной форме необходимо применить операцию логического умножения и правила минимизации (поглощения) для получения дизъюнктивной нормальной формы:

$$Y = \bigvee_{j=1}^w (k_1^j X_1 \wedge k_2^j X_2 \wedge \dots \wedge k_i^j X_i \wedge \dots \wedge k_n^j X_n), k_i^j = \{0,1\}. \quad (12)$$

Здесь представлена обобщенная запись ДНФ, где число термов в пределе равно $w = 2^n$, n – число строк в множестве (С, R) или количество переменных X в матрице Y. Все возможные решения – покрытия дефектов резервными компонентами – записываются на множестве идентификаторов строк таблицы покрытия. Если k_i^j при X_i принимает значение нуля, то X_i превращается в несущественную переменную.

4. Выбор минимальных и точных решений задачи покрытия. Процедура связана с определением конъюнктивных термов минимальной длины по Квайну в полученной ДНФ. Последующее преобразование к строкам и столбцам матрицы памяти на основе использования ранее введенного соответствия дает возможность записать минимальное покрытие или их совокупность в двумерной метрике строк и столбцов, отвечающей условиям (ограничениям) функции цели на количество резервных компонентов.

5. Реализация процедуры переназначения дефектным строкам и столбцам аналогичных исправных компонентов из резерва кристалла FPGA.

Пример 2. Выполнить процесс восстановления работоспособности матрицы блоков FPGA в части определения минимального числа резервных компонентов, покрывающих все дефекты. Матрица кристалла с дефектами и резервом, закрашенным темным цветом [19,20], представлена на рис. 7.

Матрица имеет резерв диагностирования и восстановления работоспособности дефектных ячеек, который определяется двумя строками и пятью столбцами. В соответствии с пунктом 1 модели процесса определения минимального числа резервных компонентов, покрывающих все обнаруженные дефекты в матрице памяти, строится таблица покрытия десяти дефектов $F = (F_{2,2}, F_{2,5}, F_{2,8}, F_{4,3}, F_{5,5}, F_{5,8}, F_{7,2}, F_{8,5}, F_{9,3}, F_{9,7})$ одиннадцатью строками следующей таблицы:

$X_i / F_{i,j}$	$F_{2,2}$	$F_{2,5}$	$F_{2,8}$	$F_{4,3}$	$F_{5,5}$	$F_{5,8}$	$F_{7,2}$	$F_{8,5}$	$F_{9,3}$	$F_{9,7}$
$C_2 \rightarrow X_1$	1						1			
$C_3 \rightarrow X_2$				1					1	
$C_5 \rightarrow X_3$		1			1			1		
$C_7 \rightarrow X_4$										1
$C_8 \rightarrow X_5$			1			1				
$R_2 \rightarrow X_6$	1	1	1							
$R_4 \rightarrow X_7$				1						
$R_5 \rightarrow X_8$					1	1				
$R_7 \rightarrow X_9$							1			
$R_8 \rightarrow X_{10}$								1		
$R_9 \rightarrow X_{11}$									1	1

(13)

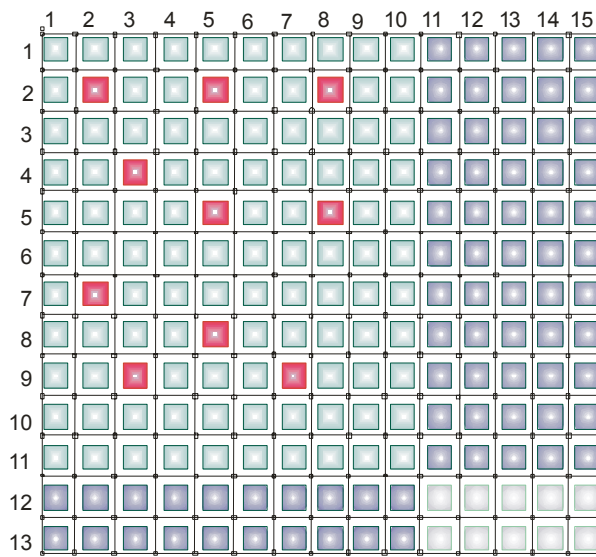


Рис. 7. Матрица блоков FPGA с дефектами и резервом

Мощность или число строк в таблице определяется конкатенацией столбцов C и строк R , находящихся во взаимно-однозначном соответствии с вектором переменных X :

$$\begin{aligned}
 C * R &= (C_2, C_3, C_5, C_7, C_8) * (R_2, R_4, R_5, R_7, R_8, R_9) \approx \\
 &\approx X = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}).
 \end{aligned}
 \tag{14}$$

По таблице покрытия выполняется построение КНФ путем записи термов по единичным значениям столбцов:

$$\begin{aligned}
 Y &= (X_1 \vee X_6)(X_3 \vee X_6)(X_5 \vee X_6)(X_2 \vee X_7)(X_3 \vee X_8) \& \\
 &\& (X_5 \vee X_8)(X_1 \vee X_9)(X_3 \vee X_{10})(X_2 \vee X_{11})(X_4 \vee X_{11}).
 \end{aligned}
 \tag{15}$$

Получение дизъюнктивной нормальной формы основывается на применении тождеств булевой алгебры, которые позволяют выполнить логическое перемножение всех десяти сомножителей и последующую минимизацию термов ДНФ путем применения оператора сограней ($ab \vee a\bar{b} = a$), аксиом поглощения и исключения одинаковых термов. Опустив промежуточные вычисления, окончательный результат приводится к следующему виду:

$$\begin{aligned}
Y = & X_1X_2X_3X_4X_5 \vee X_2X_3X_4X_5X_6X_9 \vee X_1X_2X_3X_4X_6X_8 \vee \\
& \vee X_2X_3X_4X_6X_8X_9 \vee X_1X_2X_4X_6X_8X_{10} \vee X_2X_4X_6X_8X_9X_{10} \vee \\
& \vee X_1X_4X_6X_7X_8X_{10}X_{11} \vee X_1X_2X_3X_5X_{11} \vee X_2X_3X_5X_6X_9X_{11} \vee \\
& \vee X_1X_2X_3X_6X_8X_{11} \vee X_2X_3X_6X_8X_9X_{11} \vee X_1X_2X_6X_8X_{10}X_{11} \vee \\
& \vee X_2X_6X_8X_9X_{10}X_{11} \vee X_1X_3X_5X_7X_{11} \vee X_3X_5X_6X_7X_9X_{11} \vee \\
& \vee X_1X_3X_6X_7X_8X_{11} \vee X_3X_6X_7X_8X_9X_{11} \vee X_1X_6X_7X_8X_{10}X_{11} \vee \\
& \vee X_6X_7X_8X_9X_{10}X_{11}.
\end{aligned} \tag{16}$$

Выбор термов минимальной длины, в данном случае содержащих 5 переменных, формирует множество минимальных решений:

$$Y = X_1X_2X_3X_4X_5 \vee X_1X_2X_3X_5X_{11} \vee X_1X_3X_5X_7X_{11}. \tag{17}$$

Трансформирование полученной функции к покрытию, содержащему обозначения переменных в виде строк и столбцов матрицы блоков FPGA, позволяет представить упомянутые термы в форме:

$$Y = C_2C_3C_5C_7C_8 \vee C_2C_3C_5C_8R_9 \vee C_2C_5C_8R_4R_9. \tag{18}$$

Все полученные минимальные решения удовлетворяют требованиям по ограничениям на число резервных компонентов, определенных числами:

$$(|C^r| \leq 5) \& (|R^r| \leq 2).$$

Последующая технология встроенного ремонта дефектных блоков FPGA заключается в электрическом перепрограммировании дешифратора адреса столбца или строки матрицы блоков FPGA. Применительно к FPGA, изображенной на рис. 7, переадресация столбцов с неисправными логическими блоками на резервные столбцы будет происходить, например, в соответствии с первым термом из (18), который определяет отношение:

Faulty column	2	3	5	7	8
Spare column	11	12	13	14	15

Вычислительная сложность алгебро-логического метода восстановления работоспособности в части решения задачи покрытия [17,20] определяется следующим выражением:

$$Q = 2^{|F|} + |C + R| \times 2^{|F|}, \tag{19}$$

где $2^{|F|}$ – затраты, связанные с синтезом ДНФ путем логического перемножения исключительно двухкомпонентных дизъюнкций (координата дефекта определяется номером строки и столбца), число которых равно количеству дефектных блоков; $|C + R| \times 2^{|F|}$ – верхняя граница вычислительных затрат, необходимых для минимизации полученной ДНФ на предельном множестве переменных, равном суммарному числу строк и столбцов $|C + R|$.

В худшем случае, когда координаты всех дефектных блоков по строкам и столбцам не коррелированы, – уникальны, например, диагональные дефекты, вычислительная сложность матричного метода становится зависимой только от числа дефектных ячеек, а ее аналитическая запись трансформируется к следующему виду:

$$Q = 2^{|F|} + |C + R| \times 2^{|F|} \Big|_{|C+R| \leq 2 \times |F|} = 2^{|F|} + 2 \times |F| \times 2^{|F|} = 2^{|F|} \times (1 + 2 \times |F|). \tag{20}$$

Если вместо мощности множества дефектов записать их число, равное m , тогда предыдущее выражение представляется в более простой форме:

$$Q = 2^m \times (1 + 2 \times m) = 2^m (2m + 1). \tag{21}$$

Согласно технологии встроенного сервисного обслуживания функциональных логических блоков FPGA, алгебро-логический метод восстановления работоспособности логических блоков на основе решения задачи покрытия имплементируется в кристалл, в качестве одного из компонентов инфраструктуры сервисного обслуживания (I-IP), нацеленного на поддержание работоспособности блоков FPGA и SoC в целом.

5. Заключение. Научная новизна и практическая значимость

На основе конструктивного анализа опубликованных работ существующие технологии восстановления работоспособности можно разделить на две группы. Первая ориентирована на замену единичного элемента (оптимальная и мозаичная модели). Вторая – на замену группы дефектных элементов (иерархическая структура и модель с избыточностью). С другой стороны, известные из публикаций методы можно классифицировать в три группы: 1) Дублирование логических элементов или областей кристалла, приводящее к удвоению аппаратной реализации функциональности. При фиксации неисправного элемента или области осуществляется переключение на исправный компонент с помощью мультиплексора [4]. Предложенные Xilinx модели FPGA (Field Programmable Gate Array) применимы также при ремонте компонентов FPGA от компании Altera. Основная единица измерения при ремонте – столбец или строка. 2) Использование генетических алгоритмов для диагностирования и восстановления работоспособности на основе автономной реконфигурации кристалла FPGA без использования внешних устройств управления [5]. Надежность диагностирования дефектов равна 99%, время ремонта – 36 миллисекунд вместо 660 секунд, необходимых для стандартного конфигурирования проекта. 3) Восстановление работоспособности кристаллов FPGA, не критичное ко времени, путем замены локальных CLB на избыточные запасные компоненты предложено в [6,7]. Доступный уровень объединения CLB, подлежащий замене, для критически важных приложений составляет порядка тысячи логических элементов.

Предложенные методы диагностирования: алгебро-логический и векторно-логический предлагают специалисту в области проектирования и тестирования цифровых систем на кристаллах математический аппарат, который способен осуществить диагноз дефектных компонентов на основе использования предварительно построенной таблицы неисправностей. При этом векторные решения, эффективно полученные векторно-логическим методом, можно представлять всеми возможными сочетаниями дефектов в виде термов ДНФ, что характерно для алгебро-логического метода. Второй метод эффективен при числе единиц в матрице неисправностей, превышающем 10-20%.

Алгебро-логический метод восстановления работоспособности функциональных логических блоков FPGA на основе решения задачи покрытия ориентирован на имплементацию в кристалл, в качестве одного из компонентов инфраструктуры сервисного обслуживания (I-IP), нацеленного на поддержание работоспособности блоков FPGA и SiP в целом. Метод позволяет найти точное и оптимальное решение, связанное с минимальным числом резервных блоков, необходимых для восстановления работоспособности логических блоков кристалла FPGA, при наличии дефектов любой кратности.

Технологические решения, представленные в обзорной части, а также предложенные методы диагностирования и восстановления работоспособности цифровых изделий в пакетах и на кристаллах, в части их актуальности и перспективности для рынка электронных технологий подтверждаются аналитическими исследованиями рынка электроники на 2009 год и формулируются в виде «Горячей ИТ-десятки» от Gartner Research Group: 1) Виртуализация. 2) «Облачные вычисления» (cloud computing). 3) Серверы будущего, идущие на смену blade-серверам. 4) Веб-ориентированные архитектуры. 5) Смешанные корпоративные приложения (mashups). 6) Специализированные системы. 7) Социальные сети и программное обеспечение для них. 8) Объединенные коммуникации (unified communications). 9) Бизнес-аналитика. 10) «Зеленые» ИТ. Источник: <http://www.gartner.com>.

Упомянутая выше топ-десятка достаточно хорошо коррелируется с аналитическим исследованием от компании Computer Sciences Corporation (CSC), которая оформлена в семь тенденций: 1) Новые СМИ. Internet стал полноценной структурой для создания и использования аудио-, видео- и текстового контента в масштабе планеты. 2) Социальное ПО. Социальные сети привлекают миллионы пользователей, используя общность интересов. 3) Расширенная реальность. Постепенно, но настойчиво проникает в нашу жизнь. Виртуальная реальность, в которой образы пользователей путешествуют по виртуальным мирам, становится практически целесообразной при поиске подходящих товаров, услуг, изделий без их предварительного приобретения. 4) Прозрачность информации. Позволяет видеть себя и окружающий мир с заданной степенью детализации с помощью сенсоров и

internet-камер, размещенных как в офисе, так и по всей планете. Обратная сторона медали – как спрятаться и сохранить личное пространство. 5) Инновации в беспроводной связи. Позволяют запускать любое приложение на любом устройстве в любой точке планеты. Здесь следует ожидать появления конфликтов при разделении частот между телекоммуникационными операторами, радио- и телевидением, кабельными и спутниковыми компаниями, провайдерами Internet услуг. Следует ожидать интегрированного решения данного вопроса под приоритетом беспроводных технологий с мобильными Internet-сервисами. 6) Новые платформы. Повышается уровень виртуализации. Число приложений, работающих на одной машине под управлением разных операционных систем, растет экспоненциально. «Облачные вычисления», когда пользователь платит за применение компьютерной инфраструктуры и приложений провайдерам, хранящим данные клиентов на своих серверах, существенно изменяют всю структуру вычислений. Становятся более реальными перспективы нанотехнологий, молекулярных, квантовых и оптических вычислений. Вместо кремниевых чипов будут работать более легкие и мелкие элементы: атомы, ДНК, спины электронов и свет. 7) Умный мир. Семантические и сетевые технологии позволят вычислительным устройствам интерпретировать информацию по алгоритмам естественного интеллекта, будь то текст, речь, изображения или жизненные ситуации. Компьютеры будут учить, давать рекомендации, делать предсказания на основе информации, полученной от окружающей среды и конкретного человека. Развиваются средства семантического поиска в Internet с самообучением. Создается действительно умный мир, где люди и машины будут уметь рассуждать и общаться друг с другом на основе сочетания семантических и сетевых технологий. Это приведет к появлению искусственного интеллекта, а возможно сверхинтеллекта, читающего мысли человека. Источник: <http://www.pcweek.ru>.

Упомянутые выше тенденции в развитии IT-индустрии предоставляют конструктивно мыслящим ученым пути эффективного применения творческого потенциала, который может быть востребован на рынке электронных технологий в виде новых математических, технологических и программно-аппаратных приложений. Другое применение упомянутых тенденций – верифицировать относительно них практически ориентированные научные разработки в части их валидности и рыночной привлекательности для мировой IT-индустрии.

Список литературы: 1. *Kwang-Ting (Tim) Cheng*. The Need for a SiP Design and Test Infrastructure // IEEE Design and Test of Computers. May–June, 2006. P. 181. 2. *Fontanelli A.* System-in-Package Technology: Opportunities and Challenges // Quality Electronic Design, 2008. ISQED 2008, 9th International Symposium. March, 2008. P. 589–593. 3. *Appello D., Bernardi P., Grosso M., Reorda M.S.* System-in-package testing: problems and solutions // IEEE Design & Test of Computers. Vol. 23, Issue 3. May–June, 2006. P. 203 – 211. 4. *Subhasish M.; Huang W.-J., Saxena N.R., Yu S.-Y., McCluskey E.J.* Reconfigurable architecture for autonomous self-repair // IEEE Design & Test of Computers. Volume 21, Issue 3. May–June, 2004. P. 228–240. 5. *Ross R., Hall R.* A FPGA Simulation Using Asexual Genetic Algorithms for Integrated Self-Repair // Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on Volume. Issue 15–18 June 2006. P. 301 – 304. 6. *Habermann S., Kothe R., Vierhaus H.T.* Built-in self repair by reconfiguration of FPGAs // Proceedings of the 12th IEEE International Symposium on On-Line Testing, 2006. P. 187–188. 7. *S. Pontarelli, M. Ottavi, V. Vankamamidi, A. Salsano, F. Lombardi.* Reliability Evaluation of Repairable/Reconfigurable FPGAs // 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06). October, 2006. P. 227–235. 8. *Koal T., Vierhaus H. T.* Basic Architecture for Logic Self Repair // 14th IEEE International On-Line Testing Symposium. 2008. P. 177–178. 9. *Kothe R., Vierhaus H.T., Coym T., Vermeiren W., Straube B.* Embedded Self Repair by Transistor and Gate Level Reconfiguration // IEEE Design and Diagnostics of Electronic Circuits and Systems. 2006. P. 208–213. 10. *Mange D., Sipper M., Stauffer A., Tempesti G.* Toward self-repairing and self-replicating hardware: the Embryonics approach // Evolvable Hardware, Proceedings of the Second NASA/DoD Workshop. 2000. P. 205–214. 11. *Miclea L., Szilard E., Benso A.* Intelligent agents and BIST/BISR – working together in distributed systems // Proceedings of the Test Conference. 2002. P. 940–946. 12. *Rashad S. Oreifej, Carthik A. Sharma, Ronald F. DeMara.* Expediting GA-Based Evolution Using Group Testing Techniques for Reconfigurable Hardware // Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE International Conference. Sept., 2006. P. 1–8. 13. *Anand D., Cowan B., Farnsworth O., Jakobsen P., Oakland S., Ouellette M.R., Wheeler D.L.* An on-chip self-repair calculation and fusing methodology // IEEE Design & Test of Computers. Volume 20, Issue 5. Sept.–Oct., 2003. P. 67–75. 14. *Wang S.-J., Tsai T.-M.* Test and diagnosis of faulty logic blocks in FPGAs // IEEE Proceedings Computers and Digital Techniques. Vol. 146, Issue 2. March, 1999. P. 100 – 106. 15. *Pontarelli S.,*

Cardarilli G.C., Malvoni A., Ottavi M., Re M., Salsano A. System-on-chip oriented faulttolerant sequential systems implementation methodology // IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. Oct., 2001. P. 455 – 460. **16.** *Peter Rickert, William Krenik.* Cell Phone Integration: SiP, SoC, and PoP // IEEE Design and Test of Computers. May–June, 2006. P. 188-195. **17.** *Rossen K.* Discrete Mathematics and its Applications, McGraw Hill, 2003. 824p. **18.** IEEE 1500 Web Site. <http://grouper.ieee.org/groups/1500/>. **19.** *Парфентий А.Н., Хаханов В.И., Литвинова Е.И.* Модели инфраструктуры сервисного обслуживания цифровых систем на кристаллах // АСУ и приборы автоматики. 2007. Вып. 138. С.83-99. **20.** *Hahanov V., Hahanova A., Chumachenko S., Galagan S.* Diagnosis and repair method of SoC memory / / WSEAS transactions on circuits and systems. 2008. Vol.7. P. 698-707. **21.** *Hahanov V., Obrizan V., Litvinova E., Ka Lok Man.* Algebra-logical diagnosis model for SoC F-IP // WSEAS transactions on circuits and systems. Vol. 7. 2008. P. 708-717.

Поступила в редколлегию 05.02.2009

Литвинова Евгения Ивановна, канд. техн. наук, доцент кафедры технологии и автоматизации производства РЭС и ЭВС ХНУРЭ. Научные интересы: алгоритмизация задач автоматизированного проектирования электронных вычислительных средств, автоматизация диагностирования и встроенный ремонт компонентов цифровых систем в пакете (SiP). Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-421.