

Список литературы: 1. А.С. 1126946 5G06F 5/02. Преобразователь двоично-К-ичного кода в двоичный код / А.Н. Слобожанин // Открытия, изобретения. 1984. №44. С.250. 2. А.С. 1647908 5HO3M 7/12. Преобразователь двоично-К-ичного кода в двоичный код / Н.Я.Какурин, Ю.К. Кирьяков, А.Н. Макаренко // Открытия, изобретения. 1991. № 17. С. 262-263. 3. Какурин Н.Я., Макаренко А.Н., Старчевский Д.Л. Проектирование алгоритмов функционирования преобразователей двоично-десятичных кодов последовательного типа. Часть 1. Проектирование алгоритмов преобразования // АСУ и приборы автоматики. 2005. Вып. 128. С.76-182. 4. Какурин Н.Я., Макаренко А.Н., Старчевский Д.Л. Проектирование алгоритмов функционирования преобразователей двоично-десятичных кодов последовательного типа. Часть 2. Проектирование схемных реализаций // АСУ и приборы автоматики. 2005. Вып. 131. С.167-175. 5. Бондарев В.М. Программирование на С++. Харьков. «Компания СМИТ». 2004. 284 с.

Поступила в редколлегию 02.11.2007

Какурин Николай Яковлевич, канд. техн. наук, профессор кафедры автоматизации проектирования вычислительной техники ХНУРЭ. Научные интересы: прикладная теория цифровых автоматов, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Вареца Виталий Викторович, студент группы КСС-04-2 ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Коваленко Сергей Николаевич, соискатель кафедры АПВТ ХНУРЭ. Научные интересы: цифровые датчики, устройства преобразования кодов, автоматизации проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

УДК 681.326:519.713

А.Н. ПАРФЕНТИЙ, В.И. ХАХАНОВ, Е.И. ЛИТВИНОВА

МОДЕЛИ ИНФРАСТРУКТУРЫ СЕРВИСНОГО ОБСЛУЖИВАНИЯ ЦИФРОВЫХ СИСТЕМ НА КРИСТАЛЛАХ

Предлагается алгебро-логический метод встроенного сервисного обслуживания функциональностей цифровых систем на кристаллах, ориентированный на решение задач диагностирования дефектов цифровых модулей и восстановления работоспособности матриц памяти. Метод характеризуется матричным заданием модели покрытия неисправностей тестом и дает возможность получать полное и минимальное множества дефектов в функциональностях SoC.

1. Архитектура инфраструктуры сервисного обслуживания

Сервисное обслуживание функциональных (F-IP) блоков цифровой системы на кристалле предполагает интегрированное взаимодействие технологий граничного сканирования на основе стандарта IEEE 1500 [1,2] с дополнительными нефункциональными инфраструктурными I-IP-модулями [3-6], которое гарантирует прозрачность и робастность выполнения функций SoC, технологичность изготовления, надежность и встроенный ремонт изделия в процессе его эксплуатации. Инфраструктура включает следующие I-IP-модули [5]: 1) наблюдение за состоянием внутренних линий в процессе функционирования и сервисного обслуживания; 2) тестирование функциональных модулей путем подачи проверяющих наборов; 3) диагностирование отказов и дефектов путем анализа информации, полученной на стадии тестирования; 4) восстановление работоспособности функциональных модулей при диагностировании дефектов; 5) измерение характеристик и определение параметров функционирования изделия; 6) надежность и отказоустойчивость функционирования изделия в процессе эксплуатации.

Однако реализация всех компонентов инфраструктуры не всегда оправдана экономически по причине высокой стоимости разработки. Поэтому конкретная реализация компонентов из шестерки упомянутого выше полного множества является функцией от обслуживаемой функциональности и сервисных свойств, необходимых для обеспечения работоспособности SoC. На рис. 1 предложена инфраструктура SoC [3-5], ориентированная на решение задач [7,8]: 1) моделирование одиночных дефектов; 2) построение и хранение таблицы неисправностей; 3) логический и сигнатурный анализ состояний линий; 4) безусловное

диагностирование дефектов в компонентах системы на кристалле; 5) сервисная оболочка граничного сканирования состояний линий в целях поиска дефектов; 6) встроенное диагностирование и ремонт памяти путем переадресации дефектных элементов на основе использования резервных компонентов (строк и столбцов).

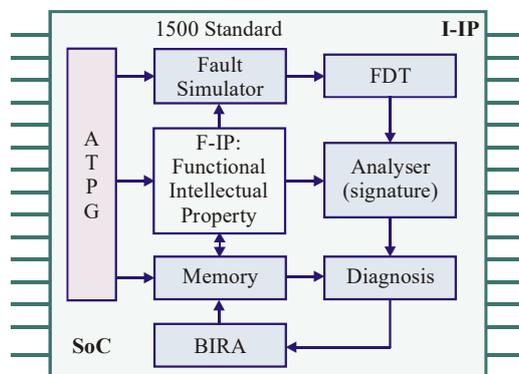


Рис. 1. Инфраструктура цифровой системы на кристалле

Все функциональности, принадлежащие инфраструктуре, реализованы в кристалле программируемой логики FPGA. Для этого используется плата аппаратного ускорения Hardware Embedded System (HES) фирмы Alatek, подключенная к PCI разъему компьютера. Модель функциональности, описанная на HDL-языке [9,10] обрабатывается программами синтеза и имплементации. Сформированный битовый файл конфигурации загружается в кристалл FPGA. Управление процессом моделирования и верификации осуществляется с помощью программы Active-HDL [www.aldec.com] и специально созданного TestBench.

Учитывая широкие функциональные и структурные возможности современных систем на кристалле, особенно возможность размещения более одного специализированного процессора для управления процедурами сервисного обслуживания, следует проектировать универсальные алгоритмы, инвариантные по отношению к программной или аппаратной реализации на кристалле. Конкретная имплементация алгоритма или метода в аппаратуру или в программу определяется требованиями заказчика, который склонен или имеет гибкую систему в целях доработки, или высокое быстродействие. Истина – всегда посередине. Поэтому далее предлагаются гибридные (программно-аппаратные) решения моделей и алгоритмов, относящихся к I-IP. Следует заметить, что аппаратные решения все более приобретают статус гибкости по отношению к быстрым изменениям функциональностей. Это связано с тем, что абсолютное доминирование памяти в масштабе кристалле предполагает размещение управляющих программ в матрице памяти. Тогда как на долю reusable logic остаются считанные проценты, ориентированные на разработку высокопроизводительных параллельных вычислителей, критичных по отношению к быстродействию.

2. Структурная модель тестирования F-IP

Модель тестирования функциональности блока SoC не может использовать только тесты для определенного класса неисправностей, как и анализаторы выходных значений. Современные технологии рассматривают данную проблему в комплексе, как тестирование и верификацию программно-аппаратного комплекса – F-IP или всей SoC, с применением механизма ассерций, функционального покрытия тестом всех маршрутов вычислительных процессов на четырех уровнях иерархии, представленных на рис. 2.

Самым нижним в данной структуре является уровень сигналов, который включает тестируемое устройство (DUT) и сигналы, соединяющие его с TestBench (драйверы и мониторы). Следующий уровень – командный (command layer). Входы DUT управляются драйвером (driver), который выполняет отдельные команды, такие как чтение и запись данных по шине. Выходы DUT наблюдаются монитором (monitor), фиксирующим изменения сигналов, на основании которых формируются команды для управления модулем контроля. Ассерции (Assertions) [11] принадлежат обоим уровням (command/signal layer), они наблюдают за отдельными сигналами и отслеживают их изменения в течение выполнения всей команды. Механизм ассерций достаточно популярен на рынке EDA для тестиро-

вания цифровых проектов на ранних стадиях проектирования, определяемых как ESL – Electronic System Level, TLM – Transaction Level Modeling [12].

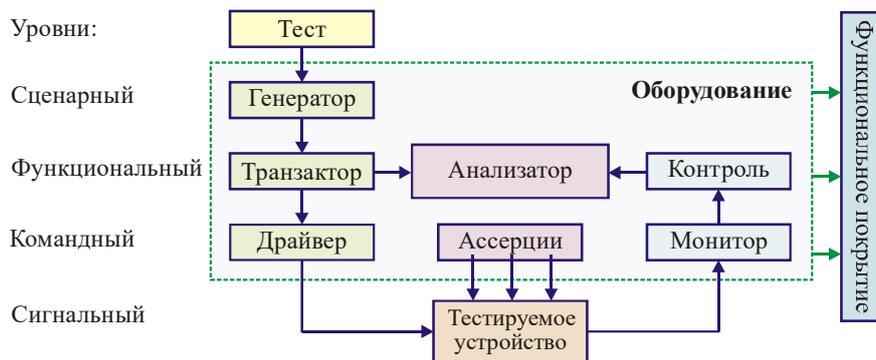


Рис. 2. Структура оборудования для тестирования F-IP

Ассерция – есть высказывание системного уровня, определяющее корректность преобразований в процессе проектирования относительно входного описания текущего этапа или требований спецификации. Механизм ассерций – есть полная система высказываний и средств их анализа, предназначенных для тестирования, верификации и валидации изделия в процессе проектирования. Процесс верификации проектируемого изделия включает создание: 1) testbench на одном из системных или HDL-языков; 2) специфицированной модели проекта, обрабатываемой в среде моделирования; 3) модели ассерций, представленной на специализированном языке или языке описания аппаратуры, помещенной в механизм (среду) для анализа ассерций. Избыточность, структурная, аппаратная и программная, создают определенные преимущества, для повышения качества проекта: 1) Применение ассерций сокращает объем кода, создаваемого непосредственно человеком для создания testbench, существенно сокращает время ручного проектирования, которое является наиболее дорогостоящим компонентом любого проекта. Сокращение объема кода в модели ассерций в несколько раз уменьшает вероятность возникновения ошибок. 2) Взаимодействие компиляторов моделирования и анализа ассерций создают условия для автоматического размещения внутри моделей устройств доступа к значениям всех внутренних сигналов. Это дает возможность оперативной идентификации места и вида дефекта и уменьшения времени обнаружения регрессий при нисходящем проектировании. 3) Сообщение о нарушении ассерции указывает место в иерархии устройства, а также момент времени, когда произошел сбой, не требуя при этом дополнительных затрат для точной локализации возникшей проблемы. 4) Момент нарушения ассерции может быть использован в среде верификации как дополнительная точка останова (breakpoint), а также для запуска специальных алгоритмов обработки, повышающих глубину диагностирования, что переводит testbench на более высокий уровень абстракции. 5) При условии, что ассерции максимально адекватно отражают спецификацию устройства, статистика процесса их моделирования используется для анализа качества покрытия функциональностей тестовыми наборами. 6) На основе практики применения ассерций наработаны специальные библиотеки верификации: OVL (Open Verification Library), обобщающие наиболее востребуемые темпоральные ограничения, которые могут найти применение при верификации широкого класса цифровых изделий.

Блок-агент (agent block), называемый транзактором (см. рис. 2), получает высокоуровневые транзакции, такие как прямое чтение и запись в память, и разбивает их на отдельные команды. Последние также посылаются в анализатор (scoreboard), который предсказывает результат транзакции. Проверочный блок сравнивает команды от монитора с полученными из scoreboard. Анализатор (scoreboard) предназначен для динамического предсказания реакции проекта путем логического моделирования укрупненной модели. Стимулы подаются на устройство и одновременно на модель, представленную функциями преобразования, уровня транзакций (TLM). Последние формируют значения сигналов, соответствующие выполнению основных операций устройства для последующего их сравнения с выходами тестируемого изделия. Блок data structure сохраняет ожидаемые реакции выхо-

дов в целях последующего сравнения с состояниями выходов DUT модуля. Функциональный слой управляется генератором сценарного уровня. Здесь формируются задания с параметрами ограничений для псевдослучайной генерации тестов. Имеется в виду совокупность генераторов, ориентированных на проверку типовых компонентов: сумматоры, шинные структуры, дешифраторы, мультиплексоры, счетчики, регистры, триггеры, память. Другой тип генераторов является перенастраиваемым в процессе тестирования. Блоки, выделенные пунктиром, создаются в начале этапа разработки. Они могут изменяться, дополняться функциями, но они никогда не меняются для отдельного конкретного теста. Именно перенастраиваемые генераторы псевдослучайных тестов задают определенную моду в процессах тестирования и верификации, ввиду своей технологичности, определенной универсальности и быстродействия. Они дают возможность на 15-20% сократить время проверки функциональности устройства. Концепция – Feedback from functional coverage to stimulus – псевдослучайный тест с обратной связью – позволяет оценить полноту покрытия функциональности тестом и скорректировать генерирование последующих наборов путем выбора других стратегий и направлений в рамках проекта. Аналогично тому, что существует coverage-driven verification, создается технология coverage-driven test generation. Подобные обратные связи используются в формальном анализе, например, в инструменте Magellan (Synopsys).

На рис. 3 представлена модель генератора тестов [13] покрытия дефектов и функциональностей, ориентированная на имплементацию в кристалл. Структура фактически представляет ядро F-IP, поскольку она решает комплекс основных задач, связанных с синтезом тестов, их верификацией, моделированием исправного поведения и неисправностей, верификацией функциональности, диагностированием дефектов в реальном масштабе времени. Отличительной особенностью представленной модели является обратная связь, влияющая на качество решения задачи тестирования в целом. Она позволяет влиять на быстродействие тестирования и на качество покрытия функциональностей (неисправностей) путем выбора тех тестовых генераторов, которые способны решить задачи покрытия еще не охваченных областей структуры (fault coverage, functional coverage) тестируемого изделия.

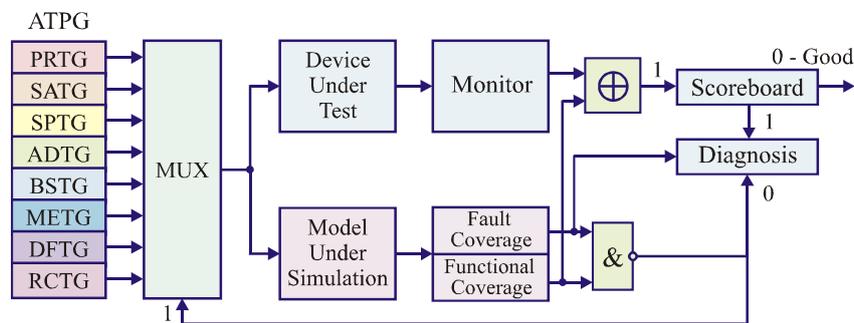


Рис. 3. Модель процесса тестирования F-IP

Особенности функционирования подсистемы тестирования заключаются в следующем:

- 1) Одновременно на функциональный блок и его модель подаются тестовые наборы, генерируемые одним из восьми синтезаторов. Модель заключена в оболочку симулятора, который ориентирован на выполнение функций: fault-free and fault simulation. Первая функция оценивает полноту покрытия функциональностей тестируемого блока, вторая – качество покрытия неисправностей.
- 2) С помощью монитор-компонента выполняется формирование реакций выходных линий и внутренних, сканированных с помощью boundary scan регистра. Состояния линий сравниваются с модельными в целях формирования вектора экспериментальной проверки, который сохраняется в модуле-анализаторе (scoreboard). Последний, при наличии всех нулевых значений, идентифицирует исправность объекта и его пригодность для выполнения функций.
- 3) Симулятор неисправностей на выходе формирует оценку качества покрытия наперед заданных дефектов и функциональностей. Если хотя бы один компонент не удовлетворяет минимальной оценке покрытия указанных свойств, осуществляется выбор другого генератора, как правило, ориентированного на генерирование тестов для еще не покрытых зон тестируемого устройства.
- 4) При наличии

управляющих сигналов от обратной связи (0) и от анализатора scoreboard (1) выполняется процедура диагностирования – определение места и вида дефекта – по информации, полученной от блока моделирования и монитора. 5) Линейка тестовых генераторов представлена восемью модулями: PRTG – псевдослучайный генератор входных стимулов с равномерным законом распределения нулей и единиц по входным переменным; SATG – тест-генератор шестнадцатиразрядных кодов на основе сигнатурного анализа; SPTG – алгоритмический генератор тест-векторов активизации одномерных путей, ориентированный на проверку заданных неисправностей; ADTG – тест-генератор, ориентированный на проверку сумматорных структур; BSTG – тестовый генератор для шинных структур передачи информации; METG – генератор тестов для матричной памяти; DFTG – генератор тестов для устройств, заданных в виде граф-схем алгоритмов; RCTG – тестовый генератор для последовательностных структур, типа счетчик, регистр, триггер.

Отличительной особенностью структуры тестирования является ее полная имплементация в кристалл, делающая ее независимой от внешних устройств, что существенно снижает стоимость сервисного обслуживания.

3. Алгебро-логическая модель диагностирования F-IP

Структура модулей сервисного обслуживания I-IP для диагностирования дефектов в функциональных блоках F-IP представлена на рис. 4. Компаратор анализирует выходные реакции модели и реального устройства на входные тестовые векторы, поступающие от генератора тестов. Несовпадения реакций формируют единичные координаты вектора экспериментальной проверки (ВЭП) для каждого входного набора. Пересечение ВЭП с таблицей неисправностей (тест-дефекты) и со схемной структурой дают множество линий и элементов, которые подозреваются дефектными на текущем тест-векторе. Пересечение подозреваемых подмножеств, определенных для каждого входного набора, дает минимальное множество подозреваемых одиночных неисправностей, мощность которого в общем случае – больше 1. Поэтому для полученного множества определяется минимальное число точек контроля, которое повышает глубину диагностирования до дефектной линии или элемента. Реакции теста в определенных ранее точках контроля сканируются с помощью boundary scan регистра, который служит в данном случае мультизондом для получения точного диагноза. Таким образом, основная интеллектуальная нагрузка приходится на scoreboard, который по сути является анализатором-контроллером, управляющим процессом поиска дефектов, после фиксации наличия неисправностей в функциональности SoC.

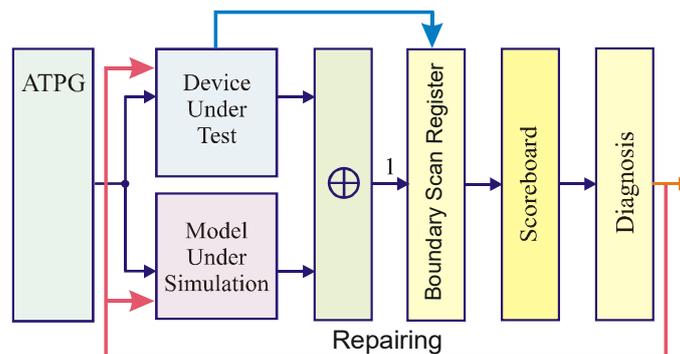


Рис. 4. Модель процесса диагностирования F-IP

Интересное решение задачи диагностирования может быть получено путем применения булевой алгебры и таблицы неисправностей M , представляющей собой декартово произведение теста T на множество заданных дефектов F , в совокупности с вектором экспериментальной проверки V , где выполнение задачи покрытия дает максимально точный результат в виде ДНФ, а каждый терм есть возможный вариант наличия в устройстве дефектов. Итак, модель процесса диагностирования представлена компонентами:

$$A = \langle T, F, M, V \rangle, T = (T_1, T_2, \dots, T_i, \dots, T_n); F = (F_1, F_2, \dots, F_j, \dots, F_m);$$

$$M = |M_{ij}|, i = \overline{1, n}; j = \overline{1, m}; V = (V_1, V_2, \dots, V_i, \dots, V_n); \{V_i, T_i, M_{ij}, F_j\} \in \{0, 1\}. \quad (1)$$

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

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

Данная формула трансформируется к ДНФ с помощью эквивалентных преобразований (операции логического умножения, минимизации (склеивания) и поглощения). В результате получается булева функция, где термы – логические произведения есть полное множество решений, представляющее собой сочетания дефектов (дающие по выходам полученный при тестировании функциональности вектор экспериментальной проверки):

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

Следующая матрица $M = T \times F$ является примером алгебро-логического анализа дефектов на основе таблицы неисправностей в функциональных блоках системы на кристалле, число которых равно 10. Тест, длиной 11 входных наборов, проверяет все введенные в таблицу неисправности. Вектор экспериментальной проверки цифрового устройства $V = (10001001001)$, полученный при выполнении диагностического эксперимента, фиксирует несовпадения выходов устройства по сравнению с моделью (золотым эталоном) на четырех (1, 5, 8 и 11) тестовых наборах.

$T_i \backslash F_j$	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

В соответствии с числом единиц (4) в векторе экспериментальной проверки V формируется количество дизъюнктивных термов КНФ (4). Каждый терм есть построчная запись дефектов (через логическую операцию ИЛИ), оказывающих влияние на искажение выходных сигналов функциональности. Далее осуществляется преобразование КНФ к ДНФ на основе правил алгебры логики, что дает возможность получить результат:

$$\begin{aligned} F &= (F_4 \vee F_{10})(F_5 \vee F_8)(F_4)(F_8 \vee F_9 \vee F_{10}) = \\ &= (F_5 \vee F_8)(F_4)(F_8 \vee F_9 \vee F_{10}) = (F_4 F_5 \vee F_4 F_8)(F_8 \vee F_9 \vee F_{10}) = \\ &= (F_4 F_5 F_8 \vee F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8 F_8 \vee F_4 F_8 F_9 \vee F_4 F_8 F_{10}) = \\ &= (F_4 F_5 F_8 \vee F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8 \vee F_4 F_8 F_9 \vee F_4 F_8 F_{10}) = \\ &= (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8). \end{aligned} \quad (4)$$

Полученный результат $F = (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8)$ содержит во всех термах дефект F_4 , означающий его обязательное присутствие в функциональности SoC. Если принять гипотезу о существовании одиночного или минимального числа кратных дефектов, то предпочтительным является решение, определяемое третьим термом $F = F_4 F_8$ – в схеме существует два дефекта, которые формируют на выходах вектор экспериментальной проверки, равный $V = (10001001001)$.

Интересный результат может дать аналогичный анализ таблицы неисправностей $M = T \times F$, выполненный по нулевым координатам вектора экспериментальной проверки:

$$F = \bigwedge_{\forall V_i=0}^{i=1, \overline{n}} \left(\bigvee_{\forall M_{ij}=1}^{j=1, \overline{m}} F_j \right). \quad (5)$$

Последующие преобразования, связанные с получением из первоначальной КНФ дизъюнктивной нормальной формы, дают следующую ДНФ:

$$\begin{aligned} \overline{F} &= (F_2 \vee F_7)(F_3 \vee F_6 \vee F_9)(F_1)(F_1 \vee F_2)(F_3)(F_5 \vee F_6)(F_7) = \\ &= F_1 F_3 F_5 F_7 \vee F_1 F_3 F_6 F_7. \end{aligned} \quad (6)$$

Анализ выражений, представленных формулами (4) и (6), приводит к интересным выводам: 1) Дефекты, которые не могут присутствовать в схеме, определяются в терминах ДНФ, полученных по нулевым строкам относительно вектора экспериментальной проверки. 2) Дефекты, которые имеются в ДНФ (6), должны быть удалены из функции (4). 3) Исключение в данном случае дефекта F_5 приводит к разрушению двух термов $F_4 F_5 F_9 \vee F_4 F_5 F_{10}$, поскольку без неисправности F_5 каждый из них в отдельности не сможет сформировать заданный вектор экспериментальной проверки. 4) Таким образом, делается единственный вывод – в схеме присутствует двукратная ошибка, определяемая термом $F = (F_4 F_8)$. 5) Вычислительная сложность получения точного и полного множества решений определяется выражением $Q = 2^{m+1}(2m+1)$. Данный метод дает возможность получить точное решение, составленное из одиночного или кратных дефектов путем решения задачи покрытия.

4. Моделирование для уточнения диагноза F-IP

Полученная дизъюнктивная форма (5) является основной моделью для поиска дефектов. Она не всегда однозначно определяет дефект функциональности, поэтому нуждается в процедурах, уточняющих диагноз. Прежде всего следует заметить, что все строки $M = T \times F$, которые отмечены нулевыми значениями вектора экспериментальной проверки, можно объединить в дизъюнкцию неисправностей, которые не могут присутствовать в схеме:

$$\begin{aligned} \overline{F} &= (F_2 \vee F_7) \vee (F_3 \vee F_6 \vee F_9) \vee (F_1) \vee (F_1 \vee F_2) \vee (F_3) \vee (F_5 \vee F_6) \vee (F_7) = \\ &= (F_2 \vee F_7 \vee F_3 \vee F_6 \vee F_9 \vee F_1 \vee F_5) = \\ &= (F_1 \vee F_2 \vee F_3 \vee F_5 \vee F_6 \vee F_7 \vee F_9). \end{aligned}$$

Обозначив отсутствие конкретной неисправности $F_i = 0$, можно сформировать входные условия для ДНФ (5) в целях последующей эмуляции (моделирования) функции при следующих начальных условиях:

$$(F_1, F_2, F_3, F_5, F_6, F_7, F_9) = (0000000).$$

Тогда результат моделирования функции $F = (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8)$ становится равным: $F = (F_4 \wedge 0 \wedge F_9 \vee F_4 \wedge 0 \wedge F_{10} \vee F_4 F_8) = F_4 F_8$.

В самом деле, если неисправности $(F_1, F_2, F_3, F_5, F_6, F_7, F_9)$, теоретически проверяемые на тестовых наборах, дают отрицательный результат – все они не искажают состояния выходов, то значит – они отсутствуют в схеме. Обоснование данного факта подтверждается следующими доказательствами.

Лемма 1. Полное множество всех возможных сочетаний дефектов, проверяемых тестом T , определяется как ДНФ, полученная преобразованием конъюнктивной формы

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

каждый терм которой записан по единичным значениям строки таблицы неисправностей $M = T \times F$, имеющей состояние ВЭП $V_i = 1$.

Исходная информация, записанная в соответствии с единичными значениями ВЭП, представляет собой полную модель неисправного поведения реального объекта, которая формирует вектор экспериментальной проверки с фиксированным числом единиц (строк таблицы неисправностей), равным k . Каждая строка формирует дизъюнкцию дефектов, записанную по ИЛИ. Число таких дизъюнкций равно k , которые логически перемножаются, образуя полное и непротиворечивое множество событий – дефектов, одновременно присутствующих в схеме. Путем перемножения элементарных дизъюнкций с последующим упрощением выражений, используя аксиомы ($a \vee ab = b$; $a \vee a = a$), получаем ДНФ, которая создает все возможные сочетания, записанные в виде элементарных конъюнкций. Ввиду тождественности выполненных преобразований полученная функция по логике эквивалентна исходной КНФ, а по сути есть технологичная форма записи всех решений – сочетаний дефектов, имеющих место быть в схеме.

Лемма 2. Все проверяемые в строках таблицы неисправностей $M = T \times F$ дефекты, отмеченные нулевыми значениями вектора экспериментальной проверки $V_i = 0$, в реальном объекте отсутствуют.

В самом деле таблица неисправностей $M = T \times F$ имеет два типа строк: единичные и нулевые относительно значения вектора экспериментальной проверки:

$$M_p(0110) \rightarrow V_p = 1; M_q(0101) \rightarrow V_q = 0.$$

Строка p выявляет наличие в схеме двух дефектов $F_2 \vee F_3$. Строка q свидетельствует о теоретической проверке неисправностей $F_2 \vee F_4$, если бы вектор $V_q = 1$. Но фактически сигнал $V_q = 0$ идентифицирует несущественность дефектов $F_2 \vee F_4$ для искажения выходов схемы. Или данные дефекты отсутствуют в тестируемом изделии. Подставив в функцию $F = F_2 \vee F_3$ нулевые сигналы для $F_2 \vee F_4$, получаем результат: $F = F_2 \vee F_3 \Big|_{F_2=F_4=0} = F_3$. Аналогично, все дефекты, которые определены в строках, имеющих нулевое значение ВЭП, будут отсутствовать в схеме. Но если они отсутствуют в схеме, то их следует исключить из ДНФ, записанной по единичным значениям ВЭП. Следовательно, имея термы ДНФ и множество дефектов, которые не могут существовать в схеме для заданного ВЭП, можно выполнить процедуру подстановки нулевых сигналов в переменные элементарных конъюнкций функции ДНФ. Но с учетом того факта, что $0 \wedge a \wedge b \wedge c \dots = 0$, результат подстановки и последующих преобразований в целях получения минимальной функции будет иметь только те термы, которые не имеют переменных – дефектов с нулевым значением сигналов. Это означает, что из ДНФ будут исключены все дефекты, относящиеся к нулевым, относительно ВЭП, строкам таблицы неисправностей.

Теорема. Минимальное множество всех возможных сочетаний дефектов, определяемых по таблице неисправностей $M = T \times F$, вычисляется путем моделирования ДНФ на множестве начальных условий

$$F = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^m k_j F_j \right) \Big|_{(V_{F_q}=0) \leftarrow (\exists M_{pq}=1) \& (V_p=0)},$$

определенных нулевыми значениями всех проверяемых дефектов, соответствующих нулевым сигналам вектора экспериментальной проверки.

В соответствии с леммой 1 полное множество всех возможных сочетаний дефектов, проверяемых тестом, определяется в виде ДНФ

$$F = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^m k_j F_j \right),$$

которая формирует все решения, удовлетворяющие единичным значениям вектора экспериментальной проверки $V_q = 1$. Оно может быть уменьшено за счет исключения тех дефектов, которые теоретически проверяются тестом, но фактически, на тестовых наборах они не искажают состояний выходов, что означает их отсутствие в реальной схеме.

Следовательно, их можно исключить из термов ДНФ, образующих полное множество всех возможных сочетаний. Механизм такого исключения, согласно лемме 2, заключается в подстановке нулевых значений переменных в термы ДНФ с последующим моделированием (упрощением) функции. Если терм имеет 0-компонент некоторой переменной F_i , то, согласно алгебре логики, весь терм обращается в 0, что означает его исключение из ДНФ. Таким образом, в результате минимизации на основе учета условий леммы 2 остается минимальная ДНФ, содержащая наименьшее число возможных сочетаний дефектов (одиночных и кратных), которое невозможно сократить без использования дополнительной диагностической информации, поступающей от мультizonда на основе boundary scan регистра.

Таким образом, предложенный алгебро-логический метод диагностирования использует булеву алгебру в качестве базового аппарата, на основе которого решается задача покрытия, путем получения дизъюнктивной формы, которая далее минимизируется путем исключения термов, имеющих переменные дефектов, относящиеся к строкам, которые идентифицированы нулевым значением ВЭП. Для незначительного числа дефектов в цифровой системе на кристалле вычислительная сложность позволяет осуществлять поиск неисправностей в реальном масштабе времени.

5. Условное диагностирование F-IP на основе ДНФ

В целях существенного уменьшения области подозреваемых дефектов на практике используется метод половинного деления, основанный на использовании интерактивной процедуры зондирования внутренних точек контроля, которые обеспечивают полученную ДНФ дефектов дополнительной информацией, которая уменьшает множество дефектов. В данном случае таким зондом может быть регистр граничного сканирования, который способен определить состояние внутренней линии в целях исключения дефектов или их подтверждения. Стратегия выбора контрольной точки ориентирована на половинное деление подозреваемого множества, – исключение на каждом шаге половины дефектов путем моделирования – упрощения исходной ДНФ. Суть метода половинного деления на дизъюнктивной нормальной форме, представляющей все возможные сочетания дефектов в схеме, можно продемонстрировать на следующем примере:

$$F = (F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8) .$$

Выбор первой точки контроля, например, $F_9 = 0$ превращает функцию в выражение:

$$F = \begin{cases} F_9 = 0 \rightarrow (F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8) = F_4F_5F_{10} \vee F_4F_8; \\ F_9 = 1 \rightarrow (F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8) = F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8. \end{cases}$$

Если же $F_9 = 1$, что означает подтверждение дефекта на линии, не уменьшает размерность ДНФ. Следовательно, необходимо ориентировать алгоритм выбора точек контроля на максимальное уменьшение исходной ДНФ после введения начальных условий ($F_j = \{0,1\}$) для моделирования. Критерием выбора точки контроля может служить взвешенность мощностей ДНФ, полученных после моделирования обоих состояний проверки.

Правила выбора контрольной точки:

Утверждение 1. Если F_j присутствует во всех термах ДНФ, то данный дефект существует обязательно в схеме, а данную линию не следует тестировать. В противном случае, если предположить, что она будет равна нулю, все термы обращаются в нуль, а это противоречит условию существования ненулевых значений вектора экспериментальной проверки V .

Утверждение 2. В схеме присутствует только одно сочетание дефектов, определенное одним термом ДНФ. Если найдено одно подтвержденное решение в виде терма ДНФ, то остальные термы следует исключить из рассмотрения путем их обращения в нуль.

Поэтому задача минимизации точек контроля сводится к выполнению двух альтернативных стратегий: 1) рассмотрение переменных в термах минимальной длины для подтверждения всех дефектов в терме путем их зондирования; 2) проверка таких переменных, которые обращают в нуль максимальное число термов ДНФ.

В случае существования функции $F = (F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8)$, которая имеет терм минимальной длины 2, а также переменную F_4 , присутствующую во всех термах, то един-

ственно лучшим решением будет проверка F_8 , которая дает при положительном результате искомое множество дефектов, а при отрицательном оставшиеся два терма, подлежащие зондированию:

$$F = \begin{cases} F_8 = 0 \rightarrow (F_4F_5F_9 \vee F_4F_5F_{10}); \\ F_8 = 1 \rightarrow (F_4F_8). \end{cases}$$

Проверка F_5 дает следующие результаты послезондового моделирования двух вариантов функций:

$$F = \begin{cases} F_5 = 0 \rightarrow F_4F_8; \\ F_5 = 1 \rightarrow F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8. \end{cases}$$

Далее, после ($F_5 = 1$), должны последовать две проверки из трех (F_9, F_{10}, F_8), которые уберают все термы, кроме одного, определяющего решение:

$$F = \begin{cases} F_9 = 0 \rightarrow F_4F_5F_{10} \vee F_4F_8; \\ F_9 = 1 \rightarrow F_4F_5F_9. \end{cases} \quad F = \begin{cases} F_{10} = 0 \rightarrow F_4F_8; \\ F_{10} = 1 \rightarrow F_4F_5F_9. \end{cases}$$

Критерием окончания процедуры диагностирования является получение одного терма ДНФ, которое идентифицирует наличие кратного дефекта в функциональности цифровой системы на кристалле.

Далее предлагается еще один пример выполнения интерактивной процедуры диагностирования на основе анализа ДНФ:

$$F = (F_3F_4F_5F_9 \vee F_2F_5F_6F_{10} \vee F_1F_4F_8F_9 \vee F_2F_4F_8F_9 \vee F_1F_4F_8F_{10}).$$

В устройстве существует кратная неисправность $F = F_1F_2F_8F_{10}$.

1) Выполняется подсчет весов каждой переменной, входящей в ДНФ:

F_i	F_1	F_2	F_3	F_4	F_5	F_6	F_8	F_9	F_{10}
$W(F_i)$	2	2	1	4	2	1	3	3	2

2) Вероятность присутствия в схеме дефектов коррелируется с их весовыми коэффициентами. Следовательно, в целях получения единственного решения в виде терма ДНФ необходимо выбирать в качестве точек контроля переменные, имеющие минимальный вес, которые будут обращать термы в нулевые составляющие. Следуя сказанному, первая и вторая точки контроля есть (F_3, F_6), имеющие минимальный вес:

$$F = (F_3F_4F_5F_9 \vee F_2F_5F_6F_{10} \vee F_1F_4F_8F_9 \vee F_2F_4F_8F_9 \vee F_1F_4F_8F_{10}) \Big|_{(F_3=0)} = \\ = F_2F_5F_6F_{10} \vee F_1F_4F_8F_9 \vee F_2F_4F_8F_9 \vee F_1F_4F_8F_{10}.$$

$$F = (F_2F_5F_6F_{10} \vee F_1F_4F_8F_9 \vee F_2F_4F_8F_9 \vee F_1F_4F_8F_{10}) \Big|_{(F_6=0)} = \\ = F_1F_4F_8F_9 \vee F_2F_4F_8F_9 \vee F_1F_4F_8F_{10}.$$

3) После каждого шага выполняется перерасчет весовых коэффициентов, который дает возможность скорректировать последующие шаги:

F_i	F_1	F_2	F_3	F_4	F_5	F_6	F_8	F_9	F_{10}
$W(F_i)$	2	1	-	3	-	-	3	2	1

Здесь установлен факт наличия в схеме дефектов (F_4, F_8), которые уже не подлежат зондированию в соответствии с условием утверждения 1.

Проверка дефекта F_2 дает следующий результат:

$$F = (F_1F_4F_8F_9 \vee F_2F_4F_8F_9 \vee F_1F_4F_8F_{10}) \Big|_{(F_2=0)} = F_1F_4F_8F_9 \vee F_1F_4F_8F_{10}.$$

Пересчет коэффициентов:

F_i	F_1	F_2	F_3	F_4	F_5	F_6	F_8	F_9	F_{10}
$W(F_i)$	2	-	-	2	-	-	2	1	1

предполагает наличие в схеме дефектов (F_1, F_4, F_8) и дополнительную проверку одной из линий (F_9, F_{10}):

$$F = (F_1F_4F_8F_9 \vee F_1F_4F_8F_{10}) \Big|_{(F_9=0)} = F_1F_4F_8F_{10}.$$

Таким образом, в результате выполнения четырех зондирований, представленных линиями (F_3, F_6, F_2, F_9), был получен точный диагноз – в схеме присутствует кратный дефект: $F = (F_1, F_4, F_8, F_{10})$.

6. Матричный метод восстановления работоспособности памяти

Память SoC в ближайшем будущем, через 5 лет, будет занимать более 90% объема кристалла [14], ориентированного на использование гибких программных средств. Актуальной представляется разработка не только моделей и методов быстрого и точного диагностирования, но и создание технологий для осуществления ремонта дефектных ячеек встроенными средствами сервисного обслуживания в реальном времени и на всех стадиях жизненного цикла изделия [15-21]. Это позволит существенно уменьшить число выводов чипа, повысить выход годной продукции, уменьшить время проектирования – time-to-market, сократить затраты на сервисное обслуживание, а также исключить внешние средства диагностирования и ремонта.

Далее предлагается оптимальный метод восстановления работоспособности памяти путем модифицированного решения задачи покрытия множества дефектных ячеек с помощью резервных элементов. Метод имеет квадратичную вычислительную сложность и может быть аппаратно реализован как в программном исполнении за пределами кристалла, так и внутри него в виде дополнительного сервисного модуля коррекции дефектов, позволяющего автоматически выполнять восстановление работоспособности элементов памяти в процессе функционирования.

Классическая задача покрытия оперирует двумя одномерными векторами (X, F), когда оператор покрытия P позволяет найти минимальное подмножество компонентов X , покрывающих своей совокупной функциональностью все элементы из F : $X_{\min} = P(X, F) \leftarrow X \cap F = X_{\min}$. Формулировка проблемы покрытия свойств одномерного вектора F двумерной матрицей $M = (C \times R)$ нуждается в приведении обоих компонентов к единой метрике – такой системе координат, которая является общим знаменателем для обеих структур. Естественно, что такой метрикой для матрицы $M = (C \times R)$ и вектора F является одномерная структура. Поэтому в данном случае априори необходимо выполнить преобразование двумерной структуры (матрицы дефектов памяти) $M = (C \times R)$ к одномерной путем выполнения конкатенации $X = (C * R)$ в целях последующего решения классической задачи покрытия путем применения формальных действий, определяемых оператором $X_{\min} = P(X, F)$.

Функция цели определяется как минимизация резервных компонентов матрицы памяти ($S - spare$), необходимых для восстановления ее работоспособности в процессе функционирования цифровой системы на кристалле путем синтеза дизъюнктивной нормальной формы покрытия дефектных элементов с последующим выбором минимального конъюнктивного термина $X^t (R^t, C^t) \in Y$, удовлетворяющего ограничениям по числу резервных строк и столбцов S_{\max}^r, S_{\max}^c , входящих в состав логического произведения:

$$Z = \min_{t=1, n} (|X^t|) \left| \left| |S^r| + |S^c| \leq S_{\max}; |S^r| \leq S_{\max}^r; |S^c| \leq S_{\max}^c \right. \right|$$

$$X^t \in Y = \{X^1, X^2, \dots, X^t, \dots, X^n\}, X^t = (X_1^t \& X_2^t \& \dots \& X_1^t \& \dots \& X_{m_t}^t),$$

где каждый результирующий конъюнктивный терм функции Y составлен из идентификаторов строк и столбцов $X^t = (R^t, C^t)$, покрывающих все дефекты в матрице памяти. Лучшее решение есть терм минимальной длины по Квайну, в котором содержатся как строки, так и столбцы, покрывающие все дефекты. В частности, решение может не содержать строки (столбцы), когда для ремонта памяти достаточно только существующих столбцов (строк) из резерва матрицы памяти. Основные пункты модели процесса определения минималь-

ного числа резервных компонентов, покрывающих все обнаруженные дефекты в матрице памяти, сводятся к следующим пунктам:

1. Преобразование двумерной модели дефектов матрицы памяти в таблицу покрытия дефектов строками и столбцами матрицы. Для достижения поставленной цели рассматривается топологическая модель памяти в виде матрицы, идентифицирующей обнаруженные дефекты:

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

Здесь координата матрицы отмечается 1, если функция исправного поведения ячейки на тесте дает единичное значение, координата идентифицируется дефектной. После фиксации всех дефектов выполняется построение таблицы покрытия дефектов $Y = |Y_{ij}|, i = \overline{1, n}; j = \overline{1, m}$, где столбцы соответствуют множеству установленных дефектов m , а строки есть номера только тех столбцов и строк матрицы памяти, которые имеют дефекты:

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

Вместо компонентов двумерной метрики C и R используется одномерный вектор, сконкатенированный из двух последовательностей C и R , мощность которого равна $n=p+q$:

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

При этом между элементами исходных наборов (C, 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 конъюнктивной нормальной формы для аналитического, полного и точного решения задачи покрытия. После формирования матрицы покрытия, содержащей нулевые и единичные координаты, выполняется синтез аналитической формы покрытия путем записи конъюнкции дизъюнкций по столбцам: число конъюнктивных термов равно количеству столбцов таблицы, а каждая дизъюнкция записывается по единичным значениям рассматриваемого столбца:

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

Из последнего выражения видно, что каждый столбец имеет только две координаты, имеющие единичное значение, а число логических произведений равно общему числу дефектов m , обнаруженных в матрице памяти.

3. Преобразование КНФ к ДНФ, дающей 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:

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

Здесь представлена обобщенная запись ДНФ, где в пределе число термов равно $w = 2^n$, n – число строк в обобщенном множестве (C, R) или количество переменных X в матрице Y , на множестве которых формируются все решения – покрытия дефектов резервными компонентами; если k_i^j при X_i принимает значение нуля, то переменная X_i превращается в несущественную.

4. Выбор минимальных и точных решений задачи покрытия. Связан с определением конъюнкций минимальной длины 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, отвечающие условиям (ограничениям) функции цели на количество резервных компонентов.

Далее предлагается иллюстрация модели процесса восстановления работоспособности матрицы памяти в части определения минимального числа резервных компонентов, покрывающих все дефекты. Матрица памяти с дефектными ячейками и резервом представлена на рис. 5 [16].

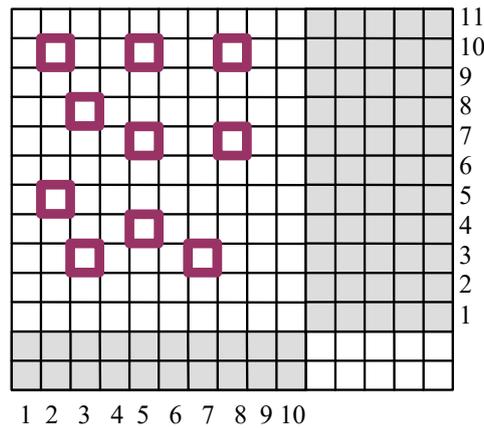


Рис. 5. Матрица памяти с дефектными ячейками и резервом

Матрица имеет ограничения на возможность диагностирования и восстановление работоспособности десяти дефектных ячеек, которые определяются 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 $F = (F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10})$ одиннадцатью строками, представленными в виде конкатенации подмножеств C и R , находящихся во взаимно-однозначном соответствии с вектором переменных X :

$$C * R = (C_2, C_3, C_5, C_7, C_8) * (R_3, R_4, R_5, R_7, R_8, R_{10}) \approx X = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}).$$

$$Y =$$

X_i / F_j	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
$C_2 \rightarrow X_1$				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		
$R_3 \rightarrow X_6$	1	1								
$R_4 \rightarrow X_7$			1							
$R_5 \rightarrow X_8$				1						
$R_7 \rightarrow X_9$					1	1				
$R_8 \rightarrow X_{10}$							1			
$R_{10} \rightarrow X_{11}$								1	1	1

В соответствии с таблицей покрытия выполняется построение КНФ, термы которой записаны по единичным значениям столбцов:

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

Последующие преобразования, связанные с получением дизъюнктивной нормальной формы, основываются на применении законов и тождеств булевой алгебры, которые позволяют выполнить логическое перемножение всех десяти сомножителей, последующую минимизацию термов ДНФ путем применения оператора сограней, аксиом поглощения, исключения одинаковых термов. Промежуточные вычисления и окончательный результат представлены в следующем виде:

$$Y = X_1 X_2 X_3 X_4 X_5 \vee X_2 X_3 X_4 X_5 X_8 X_{11} \vee X_1 X_2 X_4 X_9 X_3 X_{11} \vee \\ \vee X_1 X_3 X_2 X_4 X_9 X_{10} X_{11} \vee X_1 X_7 X_{10} X_{11} X_6 X_9 \vee X_6 X_9 X_7 X_8 X_{10} X_{11} \vee \\ \vee X_2 X_4 X_9 X_3 X_8 X_{11} \vee X_1 X_2 X_4 X_9 X_7 X_{11} \vee X_2 X_4 X_9 X_7 X_8 X_{11} \vee \\ \vee X_3 X_2 X_4 X_9 X_8 X_{10} X_{11} \vee X_1 X_2 X_4 X_9 X_7 X_{10} X_{11} \vee X_1 X_2 X_3 X_5 X_6 \vee \\ \vee X_1 X_3 X_5 X_6 X_{10} \vee X_2 X_3 X_5 X_6 X_8 X_{11} \vee X_3 X_5 X_6 X_8 X_{10} X_{11} \vee \\ \vee X_1 X_2 X_3 X_{11} X_6 X_9 \vee X_1 X_3 X_{10} X_{11} X_6 X_9 \vee X_2 X_3 X_8 X_{11} X_6 X_9 \vee \\ \vee X_1 X_2 X_7 X_{11} X_6 X_9 \vee X_2 X_7 X_8 X_{11} X_6 X_9 \vee X_3 X_8 X_{10} X_{11} X_6 X_9.$$

Выбор термов минимальной длины, содержащих 5 переменных, формирует множество оптимальных решений, имеющих вид:

$$Y = X_1 X_2 X_3 X_4 X_5 \vee X_1 X_2 X_3 X_5 X_6 \vee X_1 X_3 X_5 X_6 X_{10}.$$

Трансформирование полученной функции к покрытию, содержащему обозначения переменных в виде строк и столбцов матрицы памяти, позволяет представить решения в следующей форме:

$$Y = C_2 C_3 C_5 C_7 C_8 \vee C_2 C_3 C_5 C_8 R_3 \vee C_2 C_5 C_8 R_3 R_8.$$

Все полученные минимальные решения удовлетворяют пользователя по ограничениям на число резервных компонентов, определенных числами: $(|C^r| \leq 5) \& (|R^r| \leq 2)$. Другие решения, определенные в ДНФ, не представляют интереса, поскольку они имеют неоптимальное покрытие дефектных ячеек, определяемое числом переменных (строки + столбцы) в термах, более пяти. Последующая технология встроенного ремонта дефектных ячеек заключается в электрическом перепрограммировании дешифратора адреса столбца или строки матрицы памяти. Применительно к памяти, процедура записи или считывания информации при обращении к любой ячейке столбца 2 будет переадресована к резервному столбцу 11. Соответственно последнему полученному решению в виде первого терма ДНФ функции Y будут заменены и другие дефектные столбцы на исправные из резерва памяти: 3 – на 12; 5 – на 13; 7 – на 14; 8 – на 15.

Вычислительная сложность матричного метода восстановления работоспособности в части оптимального решения задачи покрытия определяется следующим выражением:

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

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 ячеек; $|C + R| \times 2^{|F|}$ – 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 ячеек 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:

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

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

$$Q = 2^m \times (1 + 2 \times m) = 2^m (2m + 1).$$

Согласно технологии встроенного сервисного обслуживания функциональных модулей цифровых систем на кристаллах, матричный метод восстановления работоспособности на основе решения задачи покрытия имплементируется в кристалл в качестве одного из компонентов I-IP (Infrastructure Intellectual Property), нацеленного на поддержание работоспособности матричной памяти SoC.

7. Аппаратная и программная имплементация I-IP

Инфраструктура сервисного обслуживания функциональности, представленной в SoC, включает шесть основных компонентов: 1) монитор наблюдения внутренних линий в процессе функционирования и тестирования; 2) модуль генерации проверяющих наборов; 3) блок диагностирования отказов и дефектов; 4) модуль восстановления работоспособности функциональных компонентов; 5) блок измерения параметров функционирования изделия; 6) блок обеспечения надежности и отказоустойчивости изделия. Далее рассматривается реализация первых трех блоков, ориентированных на решение следующих четырех задач: 1) снятие информации о состоянии внутренних линий с помощью boundary scan регистра; 2) моделирование исправного поведения и неисправностей для построения таблицы неисправностей в модуле генерации тестов; 3) диагностирование дефектов в цифровых компонентах на основе таблицы неисправностей; 4) восстановление работоспособности матрицы памяти цифровой системы на кристалле.

Все упомянутые компоненты сервисной инфраструктуры были имплементированы в аппаратную плату компании Aldec-Alates, которая представляет собой аппаратный акселератор программных моделей, методов и алгоритмов. Иначе, все что можно сделать программно, выполняется в аппаратном исполнении на основе языков HDL, если необходимо получить более высокое (на 2-3 порядка) быстродействие. Процесс получения аппаратного решения достаточно технологичен и представлен структурой на рис. 6. Тестовые эксперименты и верификация программно-аппаратного комплекса для реализации методов и моделей сервисного обслуживания цифровых комбинационных и последовательностных схем дают хорошие результаты увеличения быстродействия по сравнению с программными алгоритмами тестирования, моделирования и диагностирования.

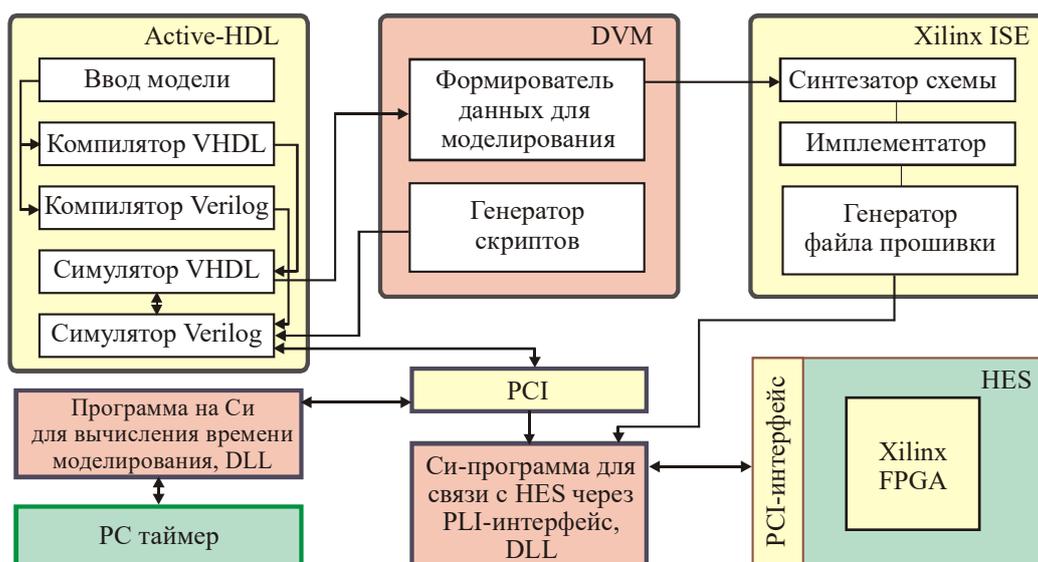


Рис. 6. Структура средств имплементации I-IP

Примеры сравнительного анализа быстродействия аппаратной (I-IP) и программной (ATPG Sigetest) реализации модулей сервисного обслуживания функциональностей DSP

SoC, реализующих компоненты программно-аппаратного комплекса, ориентированного на реализацию стандарта JPEG 2000, показаны на рис. 7.

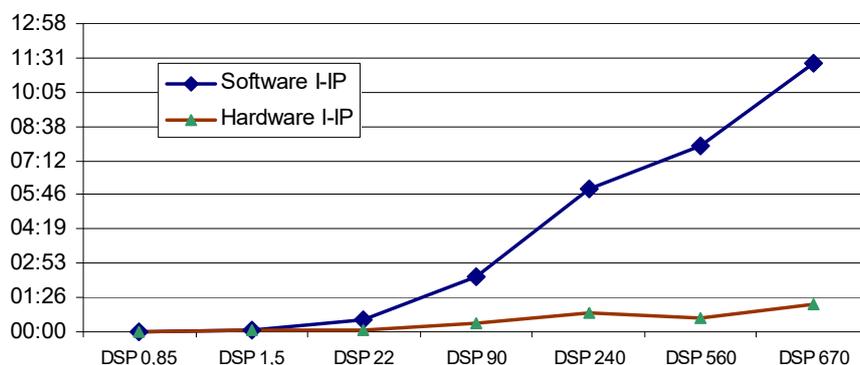


Рис. 7. Анализ быстродействия средств I-IP

Ускорение вычислительных процессов (моделирование, диагностирование) в зависимости от вентиляльной сложности проекта на кристалле может достигать одного – трех порядков. Выигрыш в быстродействии средств сервисного обслуживания более существенен для SoC большой размерности. Вычислительная сложность предложенного метода диагностирования экспоненциально зависит от числа неисправностей, одновременно присутствующих в цифровом изделии. Аппаратурные затраты на реализацию I-IP (генерация тестов, моделирование, диагностирование и ремонт) составляют не более 15% от общей функциональности F-IP, реализующей DSP SoC, которая имеет аппаратурные затраты около 1 000 000 эквивалентных вентиляей. Большая часть кристалла, порядка 80% приходится на память для хранения программ и данных, что соответствует общей тенденции к увеличению процентного отношения память – логика при реализации функциональностей в SoC.

8. Заключение

Рассмотрены вопросы аппаратной имплементации и тестовой верификации моделей, методов и алгоритмов, реализующих функциональности (генерирование тестов и моделирование неисправностей, поиск дефектов и ремонт памяти) сервисного обслуживания F-IP в силиконовом кристалле FPGA.

Научная новизна. 1) Разработана структурная и автоматная модели инфраструктуры сервисного обслуживания функциональных блоков цифровой системы на кристалле, ориентированные на решение задач генерирования тестов, моделирования неисправностей, диагностирования дефектов и восстановления работоспособности матричной памяти, которые характеризуются решением задач встроенного тестирования без использования внешних средств в реальном масштабе времени, что позволяет существенно (на 15%) повысить выход годной продукции. 2) Предложен новый матричный метод диагностирования дефектов в SoC, который характеризуется использованием дизъюнктивной нормальной формы таблицы покрытия неисправностей, что дает возможность получать полные и минимальные сочетания кратных неисправностей на основе использования процедур мультизондирования. 3) Разработан матричный метод оптимального восстановления работоспособности дефектов памяти, который отличается от аналогов применением технологии покрытия неисправностей двумерной топологией матрицы памяти, что позволяет получать минимальные и полные решения для ремонта в реальном масштабе времени, основанного на использовании резервных компонентов в виде строк и столбцов памяти.

Практическая значимость. 1) Выполнена аппаратно-программная имплементация и верификация моделей, методов и алгоритмов, реализующих функциональности (генерирование тестов и моделирование неисправностей, поиск дефектов и ремонт памяти) сервисного обслуживания F-IP в силиконовом кристалле FPGA, что дает возможность исключить внешние средства тестирования, уменьшить время диагностирования, а также повысить

выход годной продукции. 2) Метод ремонта памяти имплементирован в качестве компонента инфраструктуры сервисного обслуживания функциональных блоков цифровой системы на кристалле, что позволяет существенно (на 5-10%) повысить процент выхода годной продукции на рынке электронных технологий путем восстановления работоспособности дефектных кристаллов памяти в процессе производства и эксплуатации, а также увеличить длительность жизненного цикла матриц памяти путем восстановления их работоспособности в реальном масштабе времени. 3) Дальнейшие исследования ориентированы на разработку аппаратного модуля BIRA тестирования и восстановления работоспособности памяти при возникновении дефектов на стадии производства и эксплуатации SoC.

Список литературы: 1. *IEEE 1500 Web Site.* <http://grouper.ieee.org/groups/1500/>. 2. *IEEE P1500/D11.* January 2005. Draft Standard Testability Method for Embedded Core-based Integrated Circuits. New York. 2005. 138p. 3. *Zorian Yervant.* What is Infrastructure IP? // *IEEE Design & Test of Computers.* May-June 2002. P. 5-7. 4. *Zorian Yervant.* Advances in Infrastructure IP // *IEEE Design & Test of Computers.* May-June 2003. 49 p. 5. *Zorian Yervant and Shoukourian Samvel.* Embedded-Memory Test and Repair Infrastructure IP for SoC Yield // *IEEE Design & Test of Computers.* May-June 2003. P. 58-68. 6. *Zorian Yervant, Gizopoulos Dmytris.* Gest editors' introduction: Design for Yield and reliability // *IEEE Design & Test of Computers.* May-June 2004. P. 177-182. 7. *Abramovici M., Breuer M.A., Friedman A.D.* Digital systems testing and testable design // Computer Science Press, 1998. 652 p. 8. *Bergeron Janick.* Writing testbenches: functional verification of HDL models. – Boston: Kluwer Academic Publishers, 2001. 354 с. 9. *Хаханов В.И., Хаханова И.В.* VHDL+Verilog = синтез за минуты. Харьков: ХНУРЭ, 2006. 264 с. 10. *Хаханов В.И., Хаханова И.В.* Проектирование цифровых систем с использованием языка VHDL. Харьков: ХНУРЭ, 2003. 492 с. 11. *OpenVera Language Reference Manual: Assertions.* Version 1.4 // Synopsys Inc., 2004. 136 p. 12. *Pasricha S.* Transaction Level Modeling of SoC with SystemC 2.0 // Proc. Synopsys User Group Conference (SNUG). – San Jose: Synopsys Inc., 2002. P. 35-39. 13. *Charles E. Stroud.* A Designers Guide to Built-In Self-Test. – Kluwer Academic Publishers, 2002. 319 p. 14. *Hamdioui S., Gaydadjiev G. N., A. J. van de Goor.* The State-of-the-art and Future Trends in Testing Embedded Memories // Records IEEE International Workshop on Memory Technology, Design, and Testing, San Jose, CA, August 2004. 2004. P. 54-59. 15. *Shoukourian S., Vardanian V., Zorian Y.* SoC Yield Optimization via an Embedded-Memory Test and Repair Infrastructure // *IEEE Design and Test of Computers.* 2004. P.200-207. 16. *Memory Repair Primer – A guide to understanding embedded memory Repair options and issues,* Logic Vision. 2007. 17. *Youngs L., Paramanandam S.* Mapping and Repairing Embedded-Memory Defects // *IEEE Design and Test of Computers.* – 1997. P. 18-24. 18. *Zorian Y., Shoukourian S.* Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield // *IEEE Design and Test of Computers.* 2003. P.58-66. 19. *Ohler Ph., Hellebrand S., Wunderlich H.-J.* An Integrated Built-In Test and Repair Approach for Memories with 2D Redundancy // 12th IEEE European Test Symposium (ETS'07). 2007. P. 91-96. 20. *Choi M., Park N., Lombardi F., Kim Y.B., Piuri V.* Optimal Spare Utilization in Repairable and Reliable Memory Cores // 2003 International Workshop on Memory Technology, Design and Testing (MTDT'03). 2003. P. 64-71. 21. *Aitken Robert.* A Modular Wrapper Enabling High Speed BIST and Repair for Small Memories // *Proceedings of IEEE ITC.* April 2004. P. 997-1005.

Поступила в редколлегию 29.08.2007

Парфентий Александр Николаевич, аспирант кафедры автоматизации проектирования вычислительной техники. Научные интересы: автоматизация моделирования, верификации и тестирования систем на кристаллах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Хаханов Владимир Иванович, д-р техн. наук, профессор, декан факультета компьютерной инженерии и управления ХНУРЭ. Научные интересы: проектирование и диагностика цифровых систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-421, 70-21-326.

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, проектирование автоматизированных информационных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-486.