

УДК 681.326:519.613

**В.И. Хаханов****ИНФРАСТРУКТУРА  
ДИАГНОСТИЧЕСКОГО ОБСЛУЖИВАНИЯ SoC**

Предложена инфраструктура сервисного обслуживания функциональностей SoC, которая отличается минимальным набором процессов встроенного диагностирования в реальном масштабе времени и дает возможность осуществлять сервисы: тестирование функциональностей на основе генерируемых входных последовательностей и анализ выходных реакций; диагностирование с заданной глубиной поиска дефектов в SoC; моделирование неисправностей в целях выполнения первых двух процедур на основе таблицы неисправностей. Предложен структурно-алгебраический метод встроенного диагностирования дефектов в функциональных блоках SoC, использующий предварительный анализ таблицы неисправностей в целях уменьшения ее объема и последующих вычислений, связанных с построением дизъюнктивной нормальной формы (ДНФ), которая формирует все решения по установлению диагноза функциональностей SoC в реальном масштабе времени.

**Ключевые слова:** *диагностирование дефектов, моделирование неисправностей, диагностика в реальном времени.*

**1. Инфраструктура I-IP**

Вычислительная и аппаратная сложность современных цифровых систем на кристаллах (System-on-Chip – SoC) характеризуется миллионами эквивалентных вентиляей и требует создания и внедрения новых высокоуровневых технологий проектирования – Electronic System Level (ESL) Design, моделинга – Transaction Level Modeling (TLM) и встроенного сервисного обслуживания – Infrastructure Intellectual Property (I-IP). Это означает, что поиск быстродействующих методов и средств [1 – 17] приводит всех исследователей к необходимости повышения уровня абстракции моделей создаваемых функциональностей – Functional Intellectual Property (F-IP), встраиваемых в кристалл. Рынок программных продуктов EDA уже предлагает инструменты для автоматизации процессов моделинга и верификации устройств системного уровня, начиная с компиляторов HDL-языков (C++, SystemC, SystemVerilog, UML, SDL) [12] и заканчивая графическими оболочками (Simulink, LabView, Xilinx EDK). Данные средства позволяют создавать проекты из существующих библиотечных компонентов путем использования ESL-мэппинга и создания TLM-интерфейсов [13, 14]. Рыночная привлекательность имплементации цифровой системы в кристалл FPGA определяется применением сравнительно дешевых чипов вместо универсальных процессоров, малой потребляемой мощностью, габаритными размерами, качественным и надежным выполнением основных функций, благодаря встроенной I-IP-инфраструктуре, что является актуальным в век мобильных вычислительных устройств.

Проблема диагностирования и ремонта памяти связана с тенденцией на постоянное уменьшение площади кристалла, отводимой для оригинальной и стандартизованной логики с одновременным увеличением встроенной памяти. Как показано на рис. 1, увеличение удельного веса памяти на кристалле приводит к ее полному

доминированию для хранения данных и программ, которое к 2014 году достигнет 94% [2]. Это обеспечит не только высокое быстродействие выполнения функциональности, но и гибкость, свойственную программному продукту в части коррекции ошибок проектирования. Особенностью элементов памяти является тот факт, что в процессе их изготовления и эксплуатации отдельные ячейки под воздействием неисправностей могут выходить из состояния работоспособности. Данное обстоятельство не всегда приводит матрицу памяти к критическому состоянию, когда восстановление работоспособности невозможно. Поэтому далее рассматривается такое техническое состояние памяти, когда суммарное количество дефектных ячеек не превышает резервных возможностей изделия, предназначенных для ремонта.

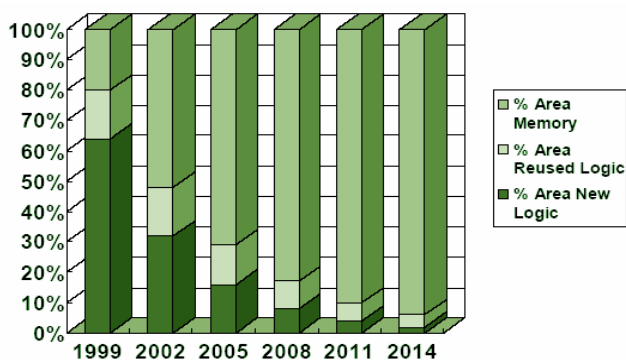


Рис. 1. Удельный вес памяти на кристалле SoC

Цель исследования – разработка алгебро-логического метода встроенного сервисного обслуживания функциональности цифровой системы на кристалле, предназначенного для диагностирования компонентов SoC и ремонта встроенной матричной памяти в реальном масштабе времени.

Задачи: 1) Анализ состояния рынка технологий сервисного обслуживания SoC; 2) Разработка алгебро-логического (AL) метода встроенного сервисного обслуживания на основе матрицы покрытия; 3) Применение AL-метода для диагностирования компонентов SoC; 4) Адаптация AL-метода для ремонта встроенной памяти; 5) Практические результаты исследования.

Современные технологии проектирования цифровых систем на кристаллах предлагают, наряду с созданием функциональных блоков F-IP, разработку сервисных модулей I-IP, ориентированных на комплексное решение проблемы качества проекта и повышение выхода годной продукции (Yield) в процессе производства, которое определяется внедрением в кристалл следующих сервисов [13]:

1) Наблюдение за состоянием внутренних и выходных линий в процессе функционирования, верификации и тестирования штатных блоков на основе использования стандарта граничного сканирования IEEE 1500 [15, 17];

2) Тестирование функциональных модулей путем подачи проверяющих наборов от различных тестовых генераторов, ориентированных на проверку дефектов или исправного поведения;

3) Диагностирование отказов и дефектов путем анализа информации, полученной на стадии тестирования и использования специальных методов встроенного поиска неисправностей на основе стандарта IEEE 1500 [15, 17];

4) Восстановление работоспособности функциональных модулей и памяти после фиксации отрицательного результата тестирования и определения места и вида дефекта при выполнении фазы диагностирования;

5) Измерение основных характеристик и параметров функционирования изделия на основе встроенных средств, позволяющих производить временные и вольт-амперные измерения;

6) Надежность и отказоустойчивость функционирования изделия в процессе эксплуатации, которая достигается диверсификацией функциональных блоков, их дублированием и восстановлением работоспособности SoC в реальном масштабе времени.

На рис. 2 представлена усеченная структура [6, 8, 11, 13], ориентированная на выполнение задач: 1) тестирования функциональностей на основе генерируемых входных последовательностей (Automated Test Pattern Generator) и анализа выходных реакций; 2) моделирования (Fault Simulator) неисправностей [9] в целях обеспечения диагностирования и ремонта на основе таблицы неисправностей (Fault Detection Table); 3) диагностирования дефектов с заданной глубиной путем использования мультитонда стандарта IEEE 1500; 4) встроенного ремонта матричной памяти на основе использования запасных компонентов (spare) [6 – 8]. Первые два пункта далее рассматриваются концептуально, а вторые два – формально составляют сущность предложенного исследования.

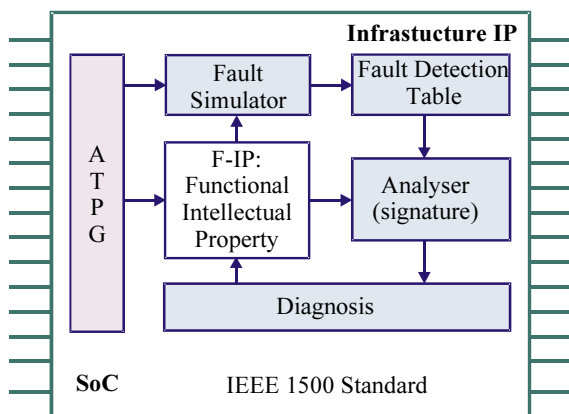


Рис. 2. Инфраструктура сервисов SoC DSP

1. Модуль синтеза тестов (см. рис.1) для проверки функциональностей и одиночных неисправностей состоит из набора генераторов входных последовательностей, которые обеспечивают создание следующих тестов [1, 2]: PRTG – псевдослучайный генератор входных стимулов с равномерным законом распределения нулевых и единичных сигналов по входным переменным; SATG – тестовый генератор шестнадцатеричных кодов на основе сигнатурного анализа; SPTG – алгоритмический генератор тестовых векторов, активизирующих одномерные логические пути, ориентированные на проверку заданных одиночных неисправностей; ADTG – тестовый генератор, предназначенный для проверки сумматорных схем АЛУ; BSTG – тестовый генератор для шинных структур приема и передачи данных; METG – генератор тестов, ориентированный на проверку матричной памяти;

DFTG – синтезатор тестов для автоматов, заданных в виде граф-схем алгоритмов; RCTG – тестовый генератор для последовательностных счетно-регистровых структур и триггерных схем.

Модуль-генератор анализирует структурно-функциональную модель блока, подлежащего тестированию, и назначает подмножество таких синтезаторов, которые обеспечивают заданное качество покрытия дефектов ( $F^c$ ) и функциональных режимов ( $P^c$ ):

$$F^c \left( \bigcup_{i=1}^{n_{\min}} T_i \right) \geq F_{\min}^c; \quad P^c \left( \bigcup_{i=1}^{n_{\min}} T_i \right) \geq P_{\min}^c, \quad (1)$$

$$T = \{T_1^{PR}, T_2^{SA}, T_3^{SP}, T_4^{AD}, T_5^{BS}, T_6^{ME}, T_7^{DF}, T_8^{RC}\}.$$

Обобщенная структура синтеза Testbench [1], представленная на рис. 3, включает также генератор HDL-кода, который предназначен для тестирования и верификации функциональностей на стадии разработки проекта.

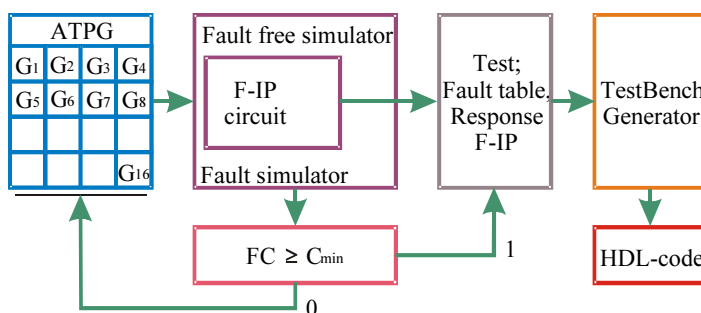


Рис. 3. Структура процесса синтеза Testbench для F-IP

Количество тестовых генераторов на стадии проектирования SoC может быть существенно большим, чем то подмножество, которое далее встраивается в кристалл. Поэтому на стадии моделирования и верификации выполняется анализ покрывающих свойств каждого тест-генератора в целях поиска их минимальной совокупной конфигурации, которая удовлетворяет выражению (1). Важно отметить, что в ближайшие 5 лет идеология синтеза тестов для цифровых систем на кристаллах будет заимствовать лучшие традиции, идущие от ESL-, TLM-проектирования [12, 16].

Это означает: 1) Использование библиотек тестов (Testbench) ведущих компаний планеты для тестирования и верификации стандартизованных функциональностей, обозначенных в качестве F-IP. 2) Применение стандартных решений сервисного обслуживания I-IP для встроенного тестирования компонентов системы на кристалле. 3) Создание собственных библиотек тестов для вновь разрабатываемых функциональностей. 4) Внедрение новой технологии синтеза тестов для цифровой системы, основанной на дискретном мэппинге [16] (рис. 4) покрытия функциональностей и дефектов исходной спецификации с помощью минимальной совокупности Testbench, из библиотеки тестов. 5) Применение встроенных средств тестопригодности, таких как IEEE boundary scan – средства граничного сканирования, и шести компонентов I-IP для повышения технологичности процедур синтеза тестов.

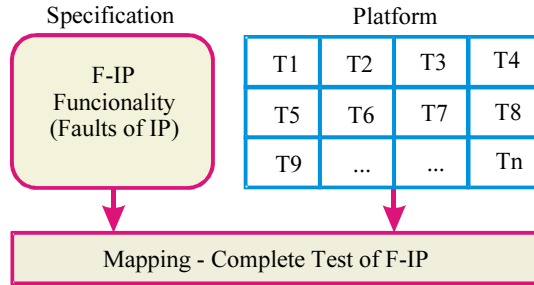


Рис. 4. Mapping модели синтеза тестов для F-IP

2. Модуль анализа неисправностей. Использует дедуктивный алгоритм, который ориентирован на проверку одиночных дефектов, генерируемых на основе аналитического или табличного описаний функциональностей SoC. Это означает, что дедуктивное моделирование может обрабатывать проекты, представленные как на вентильном, так и на любом другом, более высоком уровне абстракции (регистровом, системном). Основная идея метода заключается в создании дедуктивной модели функциональности на основе использования известного выражения [9]:

$$F = f[(X_1 \oplus T_1), (X_2 \oplus T_2), \dots, (X_j \oplus T_j), \dots, (X_{n_i} \oplus T_{n_i})] \oplus T_i, \quad (2)$$

где дедуктивная функция  $F$  на тест-векторе  $T$  есть модифицированное описание исправного поведения, позволяющее вычислять списки входных неисправностей, транспортируемые на выход схемы под воздействием входных сигналов. На примере функции Хог демонстрируется синтез дедуктивной функции по карте Карно:

$$F = \begin{array}{c|cccc} (xy) \backslash (ab) & 00 & 01 & 11 & 10 \\ \hline 00 & 0 & 1 & 0 & 1 \\ \hline 01 & 0 & 1 & 0 & 1 \\ \hline 11 & 0 & 1 & 0 & 1 \\ \hline 10 & 0 & 1 & 0 & 1 \end{array} \quad (3)$$

Переменные  $xy$  – булевы, а сигналы  $ab$  – (регистровые) для записи списков дефектов:

$$L = f(x, y, a, b) = \bar{a}b \vee a\bar{b}. \quad (4)$$

Аппаратная реализация дедуктивной функции, представленной формулой (4), изображена на рис. 5.

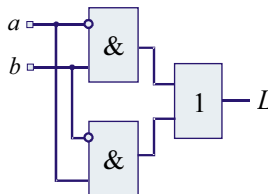


Рис. 5. Дедуктивный примитив функции Хог

Схемный примитив является универсальным по отношению к различным тестовым последовательностям. Стратегия, предложенная в работе относительно синтеза моделей, основывается на создании библиотеки дедуктивных элементов, покрывающих все стандартизованные конструктивы функциональностей, кото-

рыми оперирует разработчик, создавая в автоматизированном режиме проект в виде цифровой системы на кристалле. В данном случае речь идет о синтезе дедуктивной структуры на основе мэппинга [16], суть которого представлена на рис. 6.

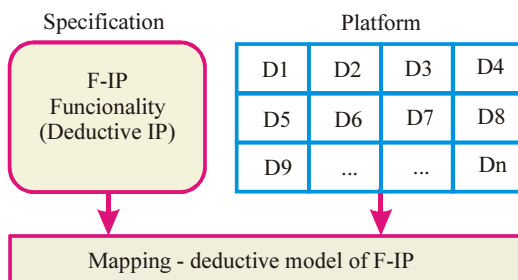


Рис. 6. Mapping дедуктивной модели для F-IP

Предложенный подход к дедуктивному анализу предполагает создание на кристалле еще одной встроенной модели, которая должна обеспечивать практически все шесть сервисов, предусмотренных стандартом инфраструктуры I-IP [13]. Платой за качество диагностического и тестового обслуживания является достаточно высокая стоимость дополнительных аппаратных затрат, которые превышают штатную функциональность в 10 – 15 раз. При этом выигрыш в быстродействии по сравнению с внешней программной реализацией дедуктивного анализа составляет 2 – 3 порядка, что практически обеспечивает сервисное обслуживание в реальном масштабе времени. Другое, более экономичное решение проблемы связано с интерактивной модификацией схемной структуры дедуктивной модели для каждого тест-вектора. Для этого используется внутренняя память кристалла, где формируется модель по правилам, определенным в (2). Мэппинг (см. рис. 6) дает дедуктивную функцию, где аппаратные затраты равны стоимости функциональности F-IP.

## 2. Алгебро-логический метод диагностирования неисправностей

Основная роль отводится технологии граничного сканирования [15,17], которая в настоящее время, внедренная в кристалл, призвана облегчать решение практически всех задач сервисного обслуживания функциональных модулей системы на кристалле. Контроллер доступа к внутренним линиям и портам регистра граничного сканирования использует ячейку или разряд регистра. В совокупности число таких ячеек, обеспечивающих в данном случае мониторинг, должно быть равно количеству проблемных наблюдаемых линий проекта, которые необходимы для точного установления диагноза. Основанная на регистре граничного сканирования процедура диагностирования использует также информацию из таблицы неисправностей (ТН), которая представляет собой множество дефектов, покрываемых тестовыми наборами. Используя информацию о результате проведения диагностического эксперимента, которая представлена в виде вектора экспериментальной проверки (ВЭП)  $V = (V_1, V_2, \dots, V_i, \dots, V_n)$ , а также таблицу [9] неисправностей  $F$ , выполняется процедура установления диагноза по выражению, записанному в форме произведения дизъюнкций всех дефектов [10], которые могут дать экспериментальную реакцию в виде  $V$ , определенном единичными и нулевыми значениями:

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

Полученная из таблицы неисправностей конъюнктивная нормальная форма трансформируется к ДНФ с помощью эквивалентных преобразований (логическое умножение, минимизация и поглощение) [10, 11]. В результате получается булева функция, где термы – логические произведения – представляют полное множество решений в виде сочетания дефектов (дающих по выходам SoC или ее компоненту вектор экспериментальной проверки  $V$ ):

$$F = \bigwedge_{\forall V_i=1}^{i=\overline{1,n}} \left( \bigvee_{\forall M_{ij}=1}^{j=\overline{1,m}} F_j \right) = \left[ \begin{array}{l} a \vee ab = b \\ a \vee a = a \end{array} \right] = \bigvee_{i=1}^{2^m} \left( \bigwedge_{j=1}^m k_j F_j \right), k_j = \{0,1\}. \quad (6)$$

Представленная процедура в общем случае диагностирует некоторое подмножество дефектов, которое в дальнейшем нуждается в уточнении путем применения дополнительного зондирования внутренних точек с помощью регистра граничного сканирования. Пример поиска дефектов рассматривается на основе следующей ниже таблицы неисправностей (столбцы – дефекты, строки – тестовые последовательности), которая является продуктом дедуктивного анализа дефектов и вектора экспериментальной проверки [18]:

$T_i / F_j$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$V$
$T_1$	1			1			1
$T_2$		1			1		1
$T_3$			1	1	1		0
$T_4$	1		1				1
$T_5$		1			1	1	1

Количество единиц в векторе экспериментальной проверки  $V$  формирует число дизъюнктивных термов КНФ (6). Каждый терм – построчная запись дефектов (через логическую операцию ИЛИ), оказывающих влияние на выходы функциональности. Уже само представление таблицы в виде аналитической записи – конъюнктивной нормальной формы – дает потенциальную возможность существенно сократить объем диагностической информации для поиска дефектов. Тем более, последующее преобразование КНФ к ДНФ на основе тождеств алгебры логики позволяет существенно уменьшить булеву функцию, что иллюстрируется следующим результатом:

$$\begin{aligned} F &= (F_1 \vee F_4)(F_2 \vee F_5)(F_3 \vee F_4 \vee F_5)(F_1 \vee F_3)(F_2 \vee F_5 \vee F_6) = \\ &= (F_1 \vee F_4)(F_2 \vee F_5)(F_3 \vee F_4 \vee F_5)(F_1 \vee F_3) = \\ &= (F_1 F_2 \vee F_2 F_4 \vee F_1 F_5 \vee F_4 F_5)(F_1 F_3 \vee F_1 F_4 \vee F_1 F_5 \vee \\ &F_3 F_3 \vee F_3 F_4 \vee F_3 F_5) = (F_1 F_2 \vee F_2 F_4 \vee F_1 F_5 \vee F_4 F_5) \\ &(F_1 F_4 \vee F_1 F_5 \vee F_3) = (F_1 F_2 F_1 F_4 \vee F_2 F_4 F_1 F_4 \vee F_1 F_5 F_1 F_4 \vee F_4 F_5 F_1 F_4) \\ &(F_1 F_2 F_1 F_5 \vee F_2 F_4 F_1 F_5 \vee F_1 F_5 F_1 F_5 \vee F_4 F_5 F_1 F_5) \\ &(F_1 F_2 F_3 \vee F_2 F_4 F_3 \vee F_1 F_5 F_3 \vee F_4 F_5 F_3) = \\ &= F_1 F_2 F_3 \vee F_2 F_3 F_4 \vee F_1 F_3 F_5 \vee F_3 F_4 F_5 \vee F_1 F_2 F_4 \vee F_1 F_4 F_5 \vee \\ &\vee F_1 F_5 = F_1 F_5 \vee F_1 F_2 F_3 \vee F_2 F_3 F_4 \vee F_3 F_4 F_5 \vee F_1 F_2 F_4. \end{aligned} \quad (7)$$

Для уменьшения количества вычислений при выполнении логического умножения в первой строке (7) исходную запись можно упростить согласно законам

булевой алгебры:

$$(F_2 \vee F_5) \wedge (F_2 \vee F_5 \vee F_6) = (F_2 \vee F_5).$$

Полученный результат  $F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4$  предоставляет все возможные решения – покрытия дефектами строк таблицы неисправностей в функциональности SoC, при условии, что вектор экспериментальной проверки имеет все единичные координаты  $V=(11111)$ . Принимая во внимание фактическое значение вектора экспериментальной проверки, равное  $V=(11011)$ , выполняется моделирование функции  $F$  путем подстановки нулевых значений дефектов, которые теоретически проверяются, но дают в векторе  $V$  нулевую координату. Такими являются дефекты:  $\bar{F} = F_3 \vee F_4 \vee F_5$ . Окончательный результат определяется следующей функцией:

$$F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4 \Big|_{F_3 \vee F_4 \vee F_5 = 0} = 0.$$

Любое сочетание – конъюнктивный терм ДНФ, приведенный в решении

$$F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4,$$

покрывает все строки таблицы неисправностей по определению, поэтому введение любой нулевой строки обязательно обращает в ноль функцию  $F$ . Поэтому корректное решение, соответствующее вектору экспериментальной проверки, должно изначально учитывать нулевые координаты вектора  $V$ . С учетом сказанного из выражения (7) на стадии записи КНФ необходимо исключить терм  $(F_3 \vee F_4 \vee F_5)$ :

$$\begin{aligned} F &= (F_1 \vee F_4)(F_2 \vee F_5)(F_1 \vee F_3)(F_2 \vee F_5 \vee F_6) = \\ &= (F_1 \vee F_4)(F_2 \vee F_5)(F_1 \vee F_3) = \\ &= (F_1F_2 \vee F_2F_4 \vee F_1F_5 \vee F_4F_5)(F_1 \vee F_3) = \\ &= F_1F_2 \vee F_1F_2F_4 \vee F_1F_5 \vee F_1F_4F_5 \vee F_1F_2F_3 \vee \\ &\vee F_2F_3F_4 \vee F_1F_3F_5 \vee F_3F_4F_5 = \\ &= F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5. \end{aligned} \tag{8}$$

Результат представляет все возможные решения, которые приводят к реакции изделия, определенного заданным вектором экспериментальной проверки:

$$F = F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5.$$

Дополнительное моделирование последней булевой функции дает окончательное решение в виде сочетания двух дефектов:

$$F = F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5 \Big|_{F_3 \vee F_4 \vee F_5 = 0} = F_1F_2.$$

Алгебро-логический метод может быть формально рассмотрен на примере следующей таблицы неисправностей  $M_1$  и представлен в виде пяти пунктов алгоритма:

$$M_1 =$$

$T_i / F_j$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$V$
$T_1$	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



1. Определение всех строк, соответствующих нулевым значениям вектора экспериментальной проверки в целях обнуления всех единичных координат найденных строк. В данном случае – это одна строка  $T_5$ .

2. Нахождение всех столбцов, которые имеют нулевые значения координат строк с нулевым состоянием ВЭП. Обнуление единичных значений найденных столбцов. В данном случае:  $F_2, F_5, F_6$ .

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

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

4. Построение конъюнктивной нормальной формы (КНФ) по единичным значениям ВЭП:

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

5. Преобразование КНФ к ДНФ с последующей минимизацией функции. В данном случае это приводит к получению искомого результата в виде сочетания неисправностей:

$$F = F_1 F_3 \vee F_1 F_4 \vee F_3 F_4.$$

Предложенный алгоритм ориентирован на предварительный анализ таблицы неисправностей в целях уменьшения объема и последующих вычислений, связанных с построением ДНФ, которая формирует все решения по установлению диагноза функциональностей SoC. Дальнейшее уточнение диагноза возможно только с применением мультizonда на основе регистра граничного сканирования данных [15].

### 3. Алгебро-логическая модель диагностирования F-IP

Структура модулей сервисного обслуживания I-IP для диагностирования дефектов в функциональных блоках F-IP представлена на рис. 7. Компаратор ( $\oplus$ ) анализирует выходные реакции модели и реального устройства на входные тестовые векторы, поступающие от генератора тестов. Несовпадения модельных и экспериментальных реакций на тесте формируют единичные координаты вектора экспериментальной проверки (ВЭП)  $V(T) = (V_1, V_2, \dots, V_i, \dots, V_n)$  для каждого входного набора. Взаимодействие ВЭП с таблицей неисправностей ( $T = [T_{tr}]$ ,  $t = \overline{1, p}$ ;  $r = \overline{1, q+n}$  размерностью  $p \times n$ ,  $p$  – число тест-векторов,  $n$  – количество разрядов boundary scan регистра) и схемной структурой дают множество линий и элементов, которые подозреваются дефектными на текущем тест-векторе.

Для организации вычислительных процессов, приводящих к точному диагнозу, чрезвычайно важна метрика или форма представления исходной информации.

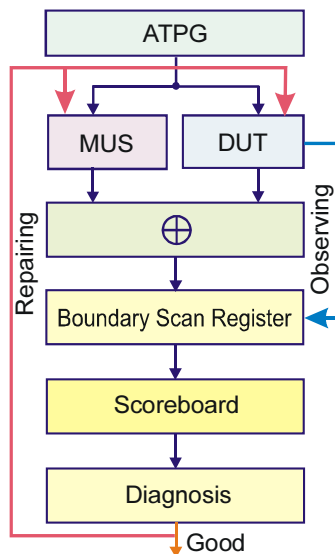


Рис. 7. Модель процесса диагностирования F-IP

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:

$$\begin{aligned}
 A &= \langle T, F, M, V \rangle, \\
 T &= (T_1, T_2, \dots, T_i, \dots, T_n); \\
 F &= (F_1, F_2, \dots, F_j, \dots, F_m); \\
 M &= |M_{ij}|, i = \overline{1, n}; j = \overline{1, m}; \\
 V &= (V_1, V_2, \dots, V_i, \dots, V_n); \\
 \{V_i, T_i, M_{ij}, F_j\} &\in \{0, 1\}.
 \end{aligned} \tag{9}$$

Решение задачи диагностирования сводится к анализу таблицы неисправностей, полученной в результате моделирования дефектов, путем записи логического произведения дизъюнкций (КНФ), записанных по единичным значениям строк таблицы неисправностей (5). Далее КНФ трансформируется к ДНФ (6) с помощью эквивалентных преобразований. В результате получается булева функция, где термы – логические произведения – есть полное множество решений, представляющее собой сочетания дефектов, дающие по выходам функциональности вектор экспериментальной проверки, полученный в результате выполнения диагностического эксперимента.

Следующая матрица  $M = T \times F$  является примером алгебро-логического анализа дефектов на основе таблицы неисправностей в функциональных блоках сис-

темы 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.

$$M = T \times F =$$

$T_i/F_j$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$V$
$T_1$				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

В соответствии с числом единиц в векторе экспериментальной проверки  $V$  формируется количество дизъюнктивных термов КНФ, равное 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:

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

Полученный результат

$$F = (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8) \tag{11}$$

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  $V = (10001001001)$ .

#### 4. Моделирование для уточнения диагноза F-IP

Полученная дизъюнктивная форма (6) 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 (6), которые не могут присутствовать в схеме:

Получение формы (5) из рассматриваемой таблицы неисправностей дает возможность определить все дефекты, которые не могут присутствовать в схеме:

$$\begin{aligned} \bar{F} &= (F_2 \vee F_7) \vee (F_3 \vee F_6 \vee F_9) \vee (F_1) \vee (F_1 \vee F_2) \vee \\ &\vee (F_3) \vee (F_5 \vee F_6) \vee (F_7) = \\ &= (F_2 \vee F_7 \vee F_3 \vee F_6 \vee F_9 \vee F_1 \vee F_5) = \\ &= (F_1 \vee F_2 \vee F_3 \vee F_5 \vee F_6 \vee F_7 \vee F_9). \end{aligned} \quad (12)$$

Анализ выражений, представленных формулами (11) и (12) 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 (14), должны быть удалены из функции (12); 3) Исключение в данном случае дефекта  $F_5$  приводит к разрушению двух термов  $F_4F_5F_9 \vee F_4F_5F_{10}$ , 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  $F = (F_4F_8)$ ; 5) Вычислительная сложность получения точного и полного множества решений определяется выражением  $Q = 2^{m+1}(2m+1)$ ,  $m$  – число дефектов.

Обозначив отсутствие конкретной неисправности  $F_i = 0$ , можно сформировать входные условия для ДНФ (11) 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  $F = (F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8)$  становится равным  $F = (F_40F_9 \vee F_40F_{10} \vee F_4F_8) = F_4F_8$ .

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

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

каждый терм которой записан по единичным значениям строки таблицы неисправностей [18]  $M = T \times F$ , 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 ( $a \vee ab = b$ ;  $a \vee a = a$ ), получается 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 был равен 1:  $V_q = 1$ . 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 это так, 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 1.** Минимальное множество всех возможных сочетаний дефектов, определяемых по таблице неисправностей  $M = T \times F$ , вычисляется путем моделирования ДНФ на множестве начальных условий

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

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 ис-

ключения 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 мультизонда на основе boundary scan-регистра.

Таким образом, предложенный алгебро-логический метод диагностирования использует булево исчисление в качестве базового аппарата для решения задачи покрытия путем получения дизъюнктивной формы, которая далее минимизируется путем исключения термов, имеющих переменные дефектов, относящиеся к строкам, с нулевыми значениями 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 [10], основанный на использовании интерактивной процедуры зондирования внутренних точек контроля, которые обеспечивают полученную ДНФ дефектов дополнительной информацией для уменьшения множества 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 примерно 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:

$$F = (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8).$$

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

$$F = \begin{cases} F_9 = 0 \rightarrow (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8) = F_4 F_5 F_{10} \vee F_4 F_8; \\ F_9 = 1 \rightarrow (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8) = F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8. \end{cases}$$

Если же  $F_9 = 1$ , что означает подтверждение дефекта на линии, то уменьшения размерности ДНФ не происходит. 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 регламентируются следующими утверждениями.

**Утверждение 1.** Если  $F_j$  присутствует во всех термах ДНФ, то данный дефект существует обязательно в схеме – его не следует тестировать. В противном случае, если предположить, что проверка будет равна нулю, то 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  $F = (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8)$ , которая имеет терм минимальной длины 2, а также переменную  $F_4$  во всех термах, единственным лучшим решением будет проверка  $F_8$ , которая дает при положительном результате искомое множество дефектов, а при отрицательном – оставшиеся два терма, подлежащие зондированию:

$$F = \begin{cases} F_8 = 0 \rightarrow (F_4 F_5 F_9 \vee F_4 F_5 F_{10}); \\ F_8 = 1 \rightarrow (F_4 F_8). \end{cases}$$

Проверка  $F_5$  дает следующие результаты послезондового моделирования двух вариантов функций:

$$F = \begin{cases} F_5 = 0 \rightarrow F_4 F_8; \\ F_5 = 1 \rightarrow F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8. \end{cases}$$

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

$$F = \begin{cases} F_9 = 0 \rightarrow F_4 F_5 F_{10} \vee F_4 F_8; \\ F_9 = 1 \rightarrow F_4 F_5 F_9. \end{cases}$$

$$F = \begin{cases} F_{10} = 0 \rightarrow F_4 F_8; \\ F_{10} = 1 \rightarrow F_4 F_5 F_9. \end{cases}$$

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:

$$F = (F_3 F_4 F_5 F_9 \vee F_2 F_5 F_6 F_{10} \vee F_1 F_4 F_8 F_9 \vee F_2 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10}).$$

В устройстве существует кратная неисправность  $F = F_1 F_2 F_8 F_{10}$ .

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:

$$F = (F_3 F_4 F_5 F_9 \vee F_2 F_5 F_6 F_{10} \vee F_1 F_4 F_8 F_9 \vee F_2 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10})|_{(F_3=0)} = F_2 F_5 F_6 F_{10} \vee F_1 F_4 F_8 F_9 \vee F_2 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10};$$

$$F = (F_2 F_5 F_6 F_{10} \vee F_1 F_4 F_8 F_9 \vee F_2 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10})|_{(F_6=0)} = F_1 F_4 F_8 F_9 \vee F_2 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10}.$$

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:

$$F = (F_1 F_4 F_8 F_9 \vee F_2 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10})|_{(F_2=0)} = F_1 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10}.$$

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  $(F_9, F_{10})$ :

$$F = (F_1 F_4 F_8 F_9 \vee F_1 F_4 F_8 F_{10})|_{(F_9=0)} = F_1 F_4 F_8 F_{10}.$$

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. Алгебро-логический метод для диагностирования и ремонта памяти

Представлен точный метод диагностирования и ремонта элементов памяти с помощью резервных элементов, который позволяет покрыть множество дефектных ячеек минимально возможным количеством избыточных компонентов. Метод ориентирован на имплементацию в инфраструктуру сервисного обслуживания для функциональности цифровой системы на кристалле. Предложены структурированные решения для реализации метода диагностирования и ремонта дефектных ячеек матрицы памяти [6, 8, 14, 19].

В процессе производства и эксплуатации любых видов памяти необходимы гарантии ее соответствия техническим условиям. Для этого предусмотрено выполнение трех процедур: 1) Тестирование памяти, заключающееся в подаче тестовых воздействий, ориентированных на выявление определенных классов дефектов [1, 5]; 2) В случае возникновения неисправности необходима дополнительная процедура диагностирования, которая позволяет определить место, причину и вид дефекта; 3) После определения множества дефектов, которые препятствуют выполнению функции памяти, необходимо активизировать процесс восстановления ра-



ботоспособности – замену дефектных элементов избыточными резервными компонентами, изначально находящимися на силиконовом кристалле [6,7]. Таким образом, упомянутые действия ориентированы на повышение выхода годных изделий (Yield) без существенных дополнительных временных и материальных затрат. Для восстановления работоспособности необходим специальный механизм ремонта памяти путем замены дефектных компонентов на исправные из резерва силиконового кристалла.

Процедура тестирования, как правило, осуществляется с помощью BIST-блока (Built-In Self Test), который представляет аппаратный быстродействующий генератор тестовых наборов, а также анализатор (сигнатурный) реакций выходов памяти на тестовые последовательности. Анализ восстановления (Repair Analysis) заключается в определении возможности покрытия дефектных элементов памяти существующими в наличии резервными компонентами. Модуль памяти представлен двумя частями: 1) функциональные ячейки, которые непосредственно применяются для хранения данных и программ при использовании модуля в системе на кристалле; 2) резервные или запасные ячейки, которые предназначены для восстановления работоспособности памяти в случае отказов функциональных ячеек. Функциональные и резервные ячейки объединяются в столбцы и строки. При обнаружении дефекта строка (столбец), содержащая дефектный элемент, отключается от функциональной структуры ячеек памяти, а на ее место подключается строка (столбец) из резерва кристалла. Поскольку количество резервных компонентов ограничено, необходим специальный механизм, позволяющий эффективно распределять ресурсы восстановления работоспособности в целях обеспечения покрытия дефектных элементов памяти минимально возможным количеством избыточных столбцов и строк.

Описанная выше процедура поиска покрытия дефектных ячеек минимальным количеством резервных строк и столбцов может быть реализована как в качестве встроенного модуля восстановления работоспособности, так и внешнего по отношению к кристаллу. Во втором случае данные об ошибках поступают извне, обрабатываются и передаются контроллеру, обеспечивающему восстановление памяти. Это приводит к значительным потерям времени. Поэтому предположение отдается on-chip реализации модуля, когда данные об ошибках передаются непосредственно из BIST. Такой механизм носит название BIRA [6, 11] (Built-In Repair Analysis) – встроенный анализ восстановления работоспособности.

Ремонт памяти осуществляется с помощью отключения дефектных элементов (строк и столбцов матрицы) путем электрического плавления перемычек и подключения резервных. Процесс пайки может быть электрическим или лазерным [9]. Устройство электрической пайки имеет меньшие габаритные размеры, чем лазерной, и применяется чаще. Пайка перемычек выполняется с помощью набора инструкций, которые могут храниться в постоянной памяти внутри чипа (hard repair) или в оперативной памяти (soft repair) [6,7,8]. Soft repair имеет ряд преимуществ: при возникновении ошибки новая исправленная инструкция может быть легко записана в память; обеспечивается экономное использование площади кристалла и достаточная надежность [4]. Hard repair позволяет использовать упрощенный производственный тест и обеспечивает обнаружение ошибок, которые в силу определенных обстоятельств не могут быть зафиксированы при soft repair, например перегрев.

Структура процессов встроенного анализа и (soft repair) самовосстановления памяти – BISR (Built-In Self Repair) – [6 – 8] представлена на рис. 8: 1) Активиза-

ция чипа, заполнение регистра BISR нулевыми значениями. 2) Запуск контроллера BIST. Тестирование памяти и накопление информации о дефектных ячейках в регистре BIRA. 3) Передача информации о дефектных ячейках в регистр BISR для последующей перепайки. 4) Сканирование BIRA-регистров, содержащих статус восстановления, контроллером BIST для получения информации о дефектах. 5) Запуск контроллера пайки в режиме записи и передача из BISR инструкций восстановления. 6) Перезагрузка чипа. Запись в регистр BISR информации о пайке перемычек, замена дефектных строк и столбцов резервными компонентами. 7) Запуск контроллера BIST в целях повторного тестирования памяти и проверки правильности результата восстановления.

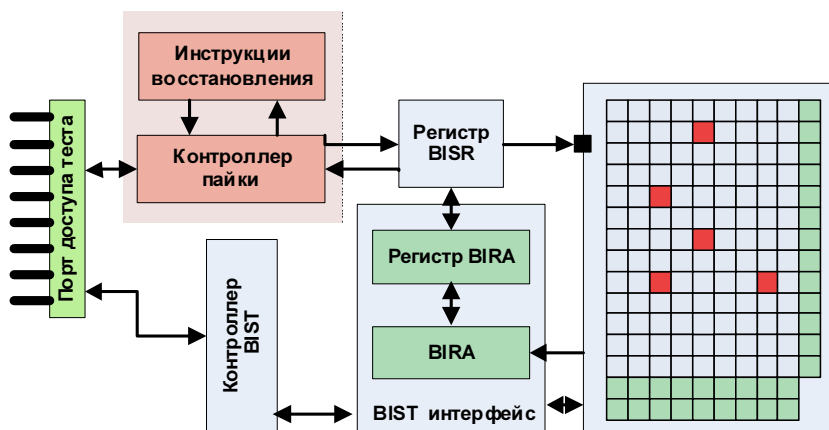


Рис. 8. Схема встроенного анализа и восстановления памяти

Функция цели  $Z$  данного исследования, исходя из современных достижений в области оперативного восстановления памяти, может быть сформулирована следующим образом: минимизация стоимости восстановления (аппаратурных затрат) модуля памяти  $M = |M_{ij}|$  в процессе эксплуатации систем на кристаллах путем использования алгебро-логического метода минимизации покрытия множества дефектных ячеек памяти системой резервных элементов в условиях ограничений  $N$  на количество последних:

$$Z = \min_i [Q_i(F)]_{Q_i(F) \leq N_{\max} = N_r + N_c},$$

где  $Q_i(F)$  – стоимость  $i$ -го варианта решения восстановления модуля памяти  $M = |M_{ij}|$  с помощью минимального подмножества строк и столбцов  $R = \{R_r, R_c\}$  резерва кристалла, покрывающего множество  $F$  дефектных ячеек памяти  $R \cap F = F, Z^* = \max |F_i|, F_i \in F \leftarrow \forall R_i$ .

Далее рассматривается метод получения минимального покрытия на примере матрицы памяти с пятью дефектными ячейками [11], двумя резервными строками и одним столбцом (см. рис. 10). Каждый резервный компонент (строка или столбец) способен восстановить работоспособность от одной до  $n$  дефектных ячеек, принадлежащих строке или столбцу.

Идея метода сводится к оптимальному замещению дефектных элементов матрицы памяти путем решения задачи покрытия дефектов-столбцов резервом строк. Для иллюстрации метода первоначально предлагается воспользоваться матрицей покрытия заданных неисправностей  $F$  некоторым количеством строк (это могут быть тестовые наборы, резервные строки)  $X$ , причем

$$|F| \geq |X| = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8\} \geq \{X_1, X_2, X_3, X_4, X_5, X_6\}.$$

Пусть задана матрица  $Y$ , имеющая вид

$$(F_j \cap X_i \neq \emptyset \rightarrow Y_{ij} = 1) \& (F_j \cap X_i = \emptyset \rightarrow Y_{ij} = 0):$$

$$Y =$$

$X_i / F_j$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$
$X_1$	1					1		
$X_2$		1	1				1	1
$X_3$					1			1
$X_4$	1			1				
$X_5$			1	1				
$X_6$					1	1		

Точное решение задачи покрытия неисправностей минимальным числом резервных строк памяти основывается на синтезе булевой функции, которая записывается как конъюнкция дизъюнкций, записанных по конstituентам единиц, соответствующих столбцам приведенной выше матрицы:

$$Y = (X_1 \vee X_4) \& (X_2) \& (X_2 \vee X_5) \& (X_4 \vee X_5) \& \\ \& (X_3 \vee X_6) \& (X_1 \vee X_6) \& (X_2) \& (X_2 \vee X_3).$$

В данном случае аналитическая запись в виде булевой функции, представленной в виде КНФ, есть исходная модель, содержащая полное множество решений задачи покрытия, которая решается путем нахождения ДНФ. Для этого выполняется процедура преобразования КНФ в ДНФ путем перемножения всех термов. В результате эквивалентных преобразований, выполненных по правилам алгебры логики, получается булева функция, содержащая все возможные покрытия неисправностей, описанные с помощью четырех вариантов комбинаций строк:

$$Y = (X_1 X_2 \vee X_2 X_4)(X_2 X_4 \vee X_4 X_5 \vee X_5 X_5 \vee X_2 X_5) \& \\ \& (X_1 X_3 \vee X_1 X_6 \vee X_6 X_6 \vee X_3 X_6)(X_2 X_2 \vee X_2 X_3) = \\ = (X_1 X_2 X_3 X_4 \vee X_2 X_4 X_6 \vee X_1 X_2 X_3 X_5 \vee X_1 X_2 X_5 X_6).$$

Минимальное решение задачи покрытия содержит всего три резервных строки, с помощью которых покрывается 8 дефектов в матрице памяти:  $Y = X_2 X_4 X_6$ .

Для использования предложенного метода восстановления работоспособности памяти необходимо иметь в виду, что каждый дефект  $F_i$  в матрице памяти принадлежит как строке, так и столбцу. Поэтому преобразование топологической модели дефектов памяти к матрице покрытия неисправностей заключается в присвоении каждому дефекту номеров строк и столбцов, которые искажаются данной неисправностью.

Для примера (рис. 9), где имеется 5 дефектных ячеек, покрываемых тремя столбцами и четверкой строк, преобразование трансформирует матрицу памяти к таблице покрытия, где левый столбец задает взаимно-однозначное соответствие

между координатами дефекта, в номерах строк и столбцов матрицы памяти и строками покрытия неисправностей:

$$Y = \begin{array}{c|ccccc} X_i / F_j & F_1 & F_2 & F_3 & F_4 & F_5 \\ \hline C_2 \rightarrow X_1 & & 1 & & 1 & \\ C_4 \rightarrow X_2 & & & 1 & & 1 \\ C_8 \rightarrow X_3 & 1 & & & & \\ R_3 \rightarrow X_4 & 1 & 1 & & & \\ R_5 \rightarrow X_5 & & & 1 & & \\ R_7 \rightarrow X_6 & & & & 1 & \\ R_{10} \rightarrow X_7 & & & & & 1 \end{array}$$

Иначе, топология матрицы памяти из двумерной метрики трансформируется в одномерную структуру строк, которые обладают определенными покрывающими свойствами относительно столбцов неисправностей.

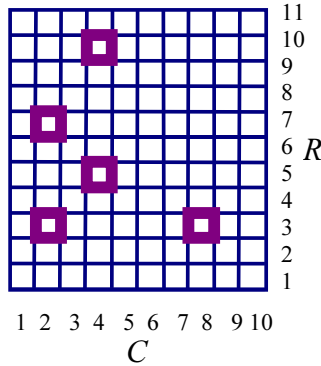


Рис. 9. Матрица памяти с дефектными ячейками

Последующая запись булевой функции формирует логическое произведение дизъюнкций, записанных по конституентам единиц, соответствующих столбцам упомянутой выше матрицы ( $F_j \cap X_i \neq \emptyset \rightarrow Y_{ij} = 1$ ):

$$\begin{aligned} Y &= (X_3 \vee X_4)(X_1 \vee X_4)(X_2 \vee X_5)(X_1 \vee X_6)(X_2 \vee X_7) = \\ &= (X_1X_3 \vee X_1X_4 \vee X_3X_4 \vee X_4)(X_1X_2 \vee X_1X_5 \vee X_2X_6 \vee \\ &\vee X_5X_6)(X_2 \vee X_7) = (X_1X_3 \vee X_4)(X_2 \vee X_7)(X_1X_2 \vee \\ &\vee X_1X_5 \vee X_2X_6 \vee X_5X_6) = (X_1X_2X_3 \vee X_2X_4 \vee X_1X_3X_7 \vee \\ &\vee X_4X_7)(X_1X_2 \vee X_1X_5 \vee X_2X_6 \vee X_5X_6) = (X_1X_2X_3 \vee \\ &\vee X_1X_2X_4 \vee X_1X_2X_3X_7 \vee X_1X_2X_4X_7 \vee X_1X_2X_3X_5 \vee \\ &\vee X_1X_2X_4X_7 \vee X_2X_4X_6 \vee X_2X_4X_5X_6 \vee X_1X_2X_3X_7 \vee \\ &\vee X_1X_3X_5X_7 \vee X_1X_2X_3X_6X_7 \vee X_1X_3X_5X_6X_7 \vee \\ &\vee X_1X_2X_4X_7 \vee X_1X_4X_5X_7 \vee X_2X_4X_6X_7 \vee X_4X_5X_6X_7) = \\ &= (X_1X_2X_3 \vee X_1X_2X_4 \vee X_2X_4X_6 \vee X_1X_3X_5X_7 \vee \\ &\vee X_1X_4X_5X_7 \vee X_4X_5X_6X_7). \end{aligned}$$

Эквивалентные преобразования позволили упростить достаточно сложную конструкцию – конъюнктивную нормальную форму – с получением минимального множества всех решений, число которых в данном случае равно шести. Подмножество минимальных решений определяется тремя конъюнктивными термами, каждый из которых содержит 3 резервных элемента для восстановления работоспособности матрицы памяти:

$$Y = X_1 X_2 X_3 \vee X_1 X_2 X_4 \vee X_2 X_4 X_6.$$

### 7. Формализация алгебро-логического метода ремонта памяти

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:

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

$$X^t \in Y = \{X^1, X^2, \dots, X^t, \dots, X^n\},$$

$$X^t = (X_1^t \& X_2^t \& \dots \& X_i^t \& \dots \& X_{m_i}^t)$$

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 строк (столбцов), когда для ремонта памяти достаточно только существующих столбцов (строк) из резерва матрицы памяти. Модель процесса определения минимального числа резервных компонентов, покрывающих все обнаруженные дефекты в матрице памяти, сводится к следующим пунктам:

1. Преобразование двумерной модели дефектов матрицы памяти в таблицу покрытия дефектов резервными 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 столбцов и строк матрицы памяти, которые имеют дефекты:

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

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 КНФ по столбцам. Здесь 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 ДНФ:

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

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 дефектами и резервом [11] представлена на рис. 10.

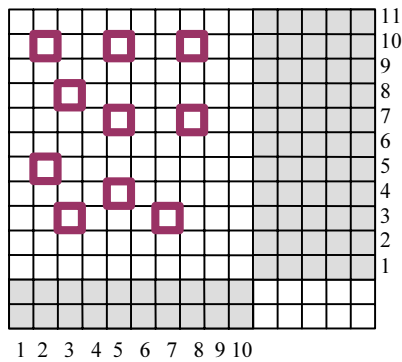


Рис. 10. Матрица памяти с дефектными ячейками и резервом

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  $X$ :

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

$X_i / F_j$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$
$C_2 \rightarrow X_1$				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 ДНФ основываются на применении законов и тождеств булевой алгебры, которые позволяют выпол-

нить логическое перемножение всех десяти сомножителей, последующую минимизацию термов ДНФ путем применения оператора сограней, аксиом поглощения, исключения одинаковых термов. Опустив промежуточные вычисления, окончательный результат представим в следующем виде:

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

Выбор термов минимальной длины, содержащих 5 переменных, формирует множество оптимальных (минимальных) 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 требованиям 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, изображенной на рис. 10, 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 алгебро-логического метода восстановления работоспособности в части 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 определяет



ся 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.

## Заключение

### 1. Диагностирование

Научная новизна и практическая значимость: 1) Предложен алгебро-логический метод и алгоритм встроенного диагностирования дефектов в функциональных блоках SoC, использующий предварительный анализ таблицы неисправностей в целях уменьшения ее объема и последующих вычислений, связанных с построением ДНФ, которая формирует все решения по установлению диагноза функциональностей SoC в реальном масштабе времени. 2) Предложена сокращенная инфраструктура сервисного обслуживания функциональностей SoC, которая отличается минимальным набором процессов встроенного диагностирования в реальном масштабе времени и дает возможность осуществлять сервисы: тестирование штатных функций на основе генерируемых входных последовательностей (Automated Test Pattern Generator) и анализ выходных реакций; диагностирование с заданной глубиной поиска дефектов путем использования мультитонда стандарта IEEE 1500; моделирование (Fault Simulation) неисправностей в целях обеспечения выполнения первых двух процедур на основе таблицы неисправностей (Fault Detection Table). 3) Предложена мэппинг-модель процесса синтеза дедуктивной структуры, которая отличается использованием библиотеки дедуктивных компонентов, покрывающих все стандартизованные конструктивы функциональностей, которыми оперирует разработчик, что дает возможность создавать в автоматизированном режиме дедуктивную модель функциональностей цифровой системы на кристалле. 4) Предложена мэппинг-модель процесса синтеза тестов, которая отличается использованием библиотеки встроенных тестовых генераторов для функциональностей DSP SoC, что дает возможность существенно уменьшить время построения тестов, предназначенных для верификации функциональностей и проверки дефектов.

## 2. Ремонт памяти

Память SoC в ближайшем будущем будет составлять более 90% объема кристалла, 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. Это позволит существенно уменьшить число выводов чипа, повысить выход годной продукции (yield), уменьшить время выхода изделия на рынок – time-to-market, сократить затраты на сервисное обслуживание, а также исключить внешние средства диагностирования и ремонта.

Алгебро-логический метод восстановления работоспособности памяти основывается на решении задачи покрытия дефектных ячеек резервными элементами путем использования аппарата булевой алгебры. 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 операции конкатенации  $X = (C * R)$  в целях последующего решения классической задачи покрытия путем применения формальных действий, определяемых оператором  $X_{\min} = P(X, F)$

Предложенный метод оптимального восстановления работоспособности дефектов памяти отличается от аналогов применением алгебро-логической технологии покрытия неисправностей двумерной топологией матрицы памяти, что позволяет получать минимальные и полные решения для последующего ремонта в реальном масштабе времени, основанного на использовании резервных компонентов в виде строк и столбцов памяти.

Практическая значимость исследования заключается в имплементации метода в инфраструктуру сервисного обслуживания функциональных блоков цифровой системы на кристалле. Это 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.

Встроенный ремонт ориентирован на все, что имеет адрес: память, мультипроцессоры, матричные процессоры. Если необходимо ремонтировать другие структуры, их следует проектировать с учетом адресуемости компонентов. Адресуемость и регулярность компонентов превращает систему в надежную, робастную, ремонтпригодную и живучую.

Алгебро-логическое представление задачи покрытия имеет привлекательность, направленную на оптимальное решение всех задач синтеза и анализа сложных систем, где существует проблема мэппинга: 1) спецификации – набором библиотечных компонентов; 2) дефектов – тестовыми последовательностями; 3) функциональностей – Testbench; 4) неисправных элементов – резервными; 5) состояний объекта – линиями наблюдения.

В целях уменьшения размерности проблемы мэппинга необходимо структурировать исходную модель путем введения иерархии, что характерно и повсеместно используется в автоматизации проектирования систем (ESL-, TLM-технологии).

Привлекательно выглядит априорное задание таблицы неисправностей в виде булевой функции с точки зрения компактности, которая на конкретном векторе экспериментальной проверки трансформируется в компактную запись, определяющую термы ДНФ, как все возможные решения неисправных компонентов, подлежащих ремонту.

Дальнейшие исследования ориентированы на разработку ремонтпригодной структуры системы и аппаратного модуля BIRA для встроенного восстановления работоспособности любых компонентов при возникновении дефектов на стадии производства и эксплуатации.

Ервант Зориан (EWDTS' 2007, Yerevan): «В настоящее время основная проблема ремонта системы на кристалле будет заключаться в разработке технологий и методов встроенного восстановления работоспособности логики, которая занимает не более 10% от площади кристалла».

#### ЛИТЕРАТУРА

1. *Bergeron J.* Writing testbenches: functional verification of HDL models. Springer, 2003.
2. *Rashinkar P., Paterson P., Singh L.* System-on-chip Verification: Methodology and Techniques, Kluwer Academic Publishers, 2002.
3. *IEEE-1800* IEEE Standard for System Verilog Language, 2005.
4. *Hamdioui S., Gaydadjiev G.N., Van de Goor A.J.* The State-of-the-art and Future Trends in Testing Embedded Memories // Records IEEE Intern. Workshop on Memory Technology, Design, and Testing, San Jose, CA, August 2004. P. 54 – 59.
5. *Zhong Y., Dropsho S.G., Shen X., Studer A., Ding C.* Miss Rate Prediction Across Program Inputs and Cache Configurations // IEEE Trans. on Computers. 2007. P. 328 – 343.
6. *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.
7. *Youngs L., Paramanandam S.* Mapping and Repairing Embedded-Memory Defects // IEEE Design and Test of Computers. 1997. P. 18 – 24.
8. *Zorian Y., Shoukourian S.* Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield // IEEE Design and Test of Computers. 2003. P. 58 – 66.
9. *Hahanov V., Kteaman H., Ghribi W., Fomina E.* HEDEFS – Hardware embedded deductive fault simulation // Proc. volume from the 3rd IFAC Workshop, Rydzyna, Poland, 2006. P. 25 – 29.
10. *Rossen K.* Discrete Mathematics and its Applications. McGraw Hill, 2003.
11. *Парфентий А.Н., Хаханов В.И., Литвинова Е.И.* Модели инфраструктуры сервисного обслуживания цифровых систем на кристаллах // АСУ и приборы автоматки. 2007. Вып. 138. С. 83 – 99.

12. Хаханов В.И., Хаханова И.В. VHDL + Verilog = Синтез за минуты. Харьков: СМИТ, 2007.
13. Zorian Y. What is Infrastructure IP // IEEE Design & Test of Computers. May-June 2002. P. 5 – 7.
14. Zorian Y., Gizopoulos D. Gest editors' introduction: Design for Yield and reliability // IEEE Design & Test of Computers. May-June 2004. P. 177 – 182.
15. IEEE 1500 Web Site. <http://grouper.ieee.org/groups/1500/>.16. Douglas Densmore, Roberto Passerone, Alberto Sangiovanni-Vincentelli. A Platform-Based taxonomy for ESL design, Design&Test of computers, september-october 2006. P. 359 – 373.
16. Francisco DaSilva, Yervant Zorian, Lee Whetsel, Karim Arabi, Rohit Kapur. Overview of the IEEE P1500 Standard // ITC International Test Conference. 2003. P. 988 – 997.
17. Автоматизация диагностирования электронных устройств / Ю.В. Малышенко и др. / Под ред. В.П. Чипулиса. М.: Энергоатомиздат, 1986.
18. Бондаренко М.Ф., Кривуля Г.Ф., Рябцев В.Г., Фрадков С.А., Хаханов В.И. Проектирование и диагностика компьютерных систем и сетей. К.: НМЦ ВО, 2000.

Статья представлена кафедрой программирования факультета прикладной математики и кибернетики Томского государственного университета и оргкомитетом 7-й Российской конференции с международным участием «Новые информационные технологии в исследовании сложных структур», поступила в научную редакцию 3 октября 2008 г.