

УДК 681.326:519.613

**В.И. ХАХАНОВ, А.В. ХАХАНОВА, Е.И. ЛИТВИНОВА**

*Харьковский национальный университет радиозлектроники, Украина*

## **АЛГЕБРО-ЛОГИЧЕСКИЙ МЕТОД РЕМОНТА ВСТРОЕННОЙ ПАМЯТИ SOC**

Предложен алгебро-логический метод оптимального восстановления работоспособности памяти, основанный на решении задачи покрытия дефектных ячеек резервными элементами путем использования аппарата булевой алгебры. Метод позволяет автоматически выполнять восстановление работоспособности элементов памяти в процессе функционирования и может иметь аппаратную или программную встроенную реализацию, представляющую собой сервисный модуль исправления дефектов.

**система на кристалле, встроенная память, дефектный элемент, восстановление работоспособности**

### **Введение**

Вычислительная и аппаратная сложность современных цифровых систем на кристаллах (System-on-Chip – SoC) характеризуется миллионами эквивалентных вентилях 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 кри-

сталл 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 работоспособ-

ности невозможно. Поэтому далее рассматривается такое техническое состояние памяти, когда суммарное количество дефектных ячеек не превышает резервных возможностей изделия, предназначенных для ремонта.

Цель исследования – разработка алгебро-логического метода для диагностирования и ремонта встроенной матричной памяти в реальном масштабе времени.

Задачи: 1) анализ технологии сервисного обслуживания SoC; 2) разработка метода встроенного сервисного обслуживания на основе матрицы покрытия; 3) формализация алгебро-логического AL-метода для ремонта встроенной памяти.

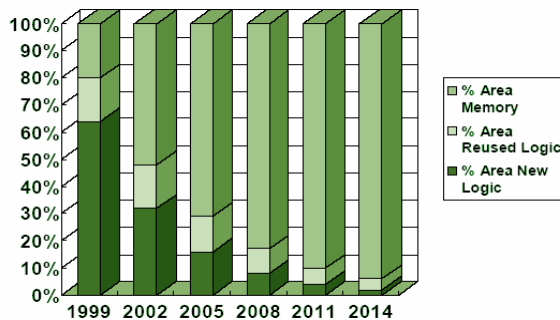


Рис. 1. Удельный вес памяти на кристалле SoC

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) диагностирование отказов и дефектов путем анализа информации, полученной на стадии тестирования и использования специальных методов встроенного поиска неисправностей на основе стандарта IEEE 1500 [9,15,17];

2) восстановление работоспособности функциональных модулей и памяти после фиксации отрицательного результата тестирования и опре-

деления места и вида дефекта при выполнении фазы диагностирования;

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

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

### Метод 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 - 9, 14, 18].

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, 8]; 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 работоспособности – замену дефектных элементов избыточными резервными компонентами, изначально находящимися на силиконо-

вом 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 предпочтение 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. 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 - 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. 2:

- 1) Активизация 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 повторно тестирования памяти и проверки правильности результата восстановления.

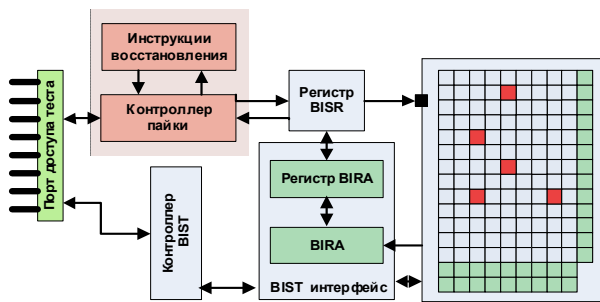


Рис. 2. Схема встроенного анализа и восстановления памяти

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 (рис. 3). 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

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

$$Y =$$

| $X_i \backslash F_j$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ |
|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| $X_1$                | 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 конститuentам 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 конъюнктивной нормальной формы (КНФ), есть исходная модель, содержащая полное множество решений задачи покрытия, которая решается путем нахождения дизъюнктивной нормальной формы (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.3), 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:

$$Y = \begin{array}{c|ccccc} X_i / F_j & F_1 & F_2 & F_3 & F_4 & F_5 \\ \hline \tilde{N}_2 \rightarrow X_1 & & 1 & & 1 & \\ \tilde{N}_4 \rightarrow X_2 & & & 1 & & 1 \\ \tilde{N}_8 \rightarrow X_3 & 1 & & & & \\ R_3 \rightarrow X_4 & 1 & 1 & & & \\ R_5 \rightarrow X_5 & & & 1 & & \\ R_7 \rightarrow X_6 & & & & 1 & \\ R_{10} \rightarrow X_7 & & & & & 1 \end{array}$$

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

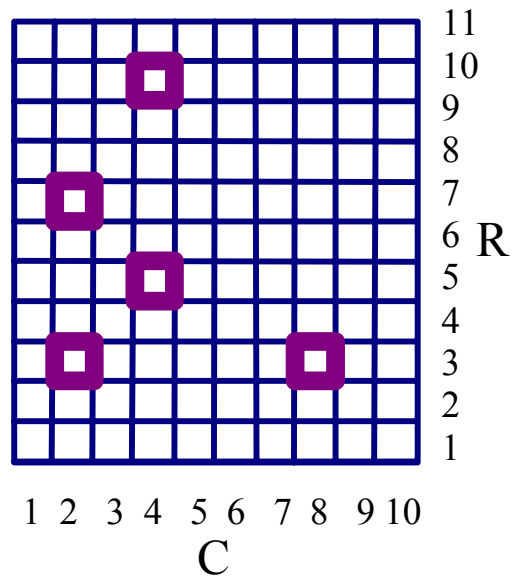


Рис. 3. Матрица памяти с дефектными ячейками

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

$$\begin{aligned}
Y &= (X_3 \vee X_4)(X_1 \vee X_4)(X_2 \vee X_5)(X_1 \vee X_6)(X_2 \vee X_7) = \\
&= (X_1 X_3 \vee X_1 X_4 \vee X_3 X_4 \vee X_4)(X_1 X_2 \vee X_1 X_5 \vee X_2 X_6 \vee \\
&\vee X_5 X_6)(X_2 \vee X_7) = (X_1 X_3 \vee X_4)(X_2 \vee X_7)(X_1 X_2 \vee \\
&\vee X_1 X_5 \vee X_2 X_6 \vee X_5 X_6) = (X_1 X_2 X_3 \vee X_2 X_4 \vee X_1 X_3 X_7 \vee \\
&\vee X_4 X_7)(X_1 X_2 \vee X_1 X_5 \vee X_2 X_6 \vee X_5 X_6) = (X_1 X_2 X_3 \vee \\
&\vee X_1 X_2 X_4 \vee X_1 X_2 X_3 X_7 \vee X_1 X_2 X_4 X_7 \vee X_1 X_2 X_3 X_5 \vee \\
&\vee X_1 X_2 X_4 X_7 \vee X_2 X_4 X_6 \vee X_2 X_4 X_5 X_6 \vee X_1 X_2 X_3 X_7 \vee \\
&\vee X_1 X_3 X_5 X_7 \vee X_1 X_2 X_3 X_6 X_7 \vee X_1 X_3 X_5 X_6 X_7 \vee \\
&\vee X_1 X_2 X_4 X_7 \vee X_1 X_4 X_5 X_7 \vee X_2 X_4 X_6 X_7 \vee X_4 X_5 X_6 X_7 = \\
&= (X_1 X_2 X_3 \vee X_1 X_2 X_4 \vee X_2 X_4 X_6 \vee X_1 X_3 X_5 X_7 \vee \\
&\vee X_1 X_4 X_5 X_7 \vee X_4 X_5 X_6 X_7).
\end{aligned}$$

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, каждое из которых содержит 3 резервных элемента для восстановления работоспособности матрицы памяти:

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

### Формализация алгебро-логического метода ремонта памяти

Функция цели определяется как минимизация резервных компонентов матрицы памяти ( $S - \text{spare}$ ), 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:

$$\begin{aligned}
Z &= \min_{t=1, n} \left( |X^t| \right) \left| \left| S^r \right| + \left| S^c \right| \leq S_{\max}^r; \left| S^r \right| \leq S_{\max}^r; \left| S^c \right| \leq S_{\max}^c \right. \\
X^t &\in Y = \{X^1, X^2, \dots, X^t, \dots, X^n\}, \\
X^t &= (X_1^t \& X_2^t \& \dots \& X_i^t \& \dots \& X_m^t)
\end{aligned}$$

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  $n = p + q$ :

$$X = C * R = (C_1, C_2, \dots, C_i, \dots, C_p) * (R_1, R_2, \dots, R_j, \dots, R_q) =$$

$$= X^c * X^r = (X_1, X_2, \dots, X_i, \dots, X_p, X_{p+1}, X_{p+2}, \dots, X_{p+j}, \dots, X_{p+q}).$$

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 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  $k_i^j = 0$  то  $X_i$  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. 4.

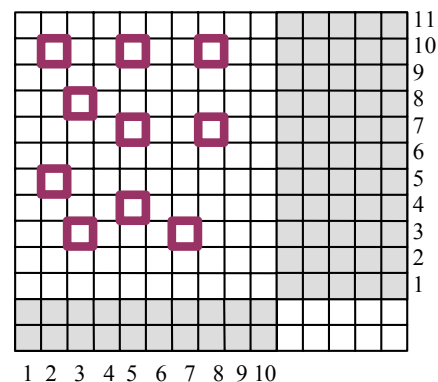


Рис. 4. Матрица памяти с дефектными ячейками и резервом

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

$$F = (F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10})$$

одинадцатью строками, представленными в виде конкатенации подмножеств С и R, находящихся во взаимно-однозначном соответствии с вектором переменных X:

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

| $X_j / F_j$                   | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ | $F_9$ | $F_{10}$ |
|-------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| $\tilde{N}_2 \rightarrow X_1$ |       |       |       | 1     |       |       |       |       |       | 1        |
| $\tilde{N}_3 \rightarrow X_2$ |       | 1     |       |       |       |       | 1     |       |       |          |
| $\tilde{N}_5 \rightarrow X_3$ |       |       | 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        |

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. Опустив промежуточные вычисления, оконча-

тельный результат представлен в следующем виде:

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

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:

$$Y = \tilde{N}_2 \tilde{N}_3 \tilde{N}_5 \tilde{N}_7 \tilde{N}_8 \vee \tilde{N}_2 \tilde{N}_3 \tilde{N}_5 \tilde{N}_8 R_3 \vee C_2 C_5 C_8 R_3 R_8.$$

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

$$(|\tilde{N}^r| \leq 5) \& (|R^r| \leq 2).$$

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.4, 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 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:

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

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 кри-

сталл, в качестве одного из компонентов I-IP (Infrastructure Intellectual Property), нацеленного на поддержание работоспособности матричной памяти SoC.

## Выводы

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

Ф 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 алгебрологической 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.

Дальнейшие исследования ориентированы на разработку ремонтпригодной структуры системы и аппаратного модуля BIRA для встроенного восстановления работоспособности памяти 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. – 512 p.
2. Rashinkar P., Paterson P., Singh L. System-on-chip Verification: Methodology and Techniques.– Kluwer Academic Publishers, 2002.– 393 p.
3. IEEE-1800 IEEE Standard for System Verilog Language. – 2005. – 586 p.
4. S. Hamdioui, G. N. Gaydadjiev, 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. – P. 54-59.
5. Zorian Y. Today's SoC Test Challenges // ITC International Test Conference.– 2005.
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. Zorian Y., Yessayan A. IEEE 1500 Utilization in SoC Design and Test // ITC International Test Conference.– 2005.
10. Rossen K. Discrete Mathematics and its Applications. – McGraw Hill, 2003. – 824 p.
11. Парфентий А.Н., Хаханов В.И., Литвинова Е.И. Модели инфраструктуры сервисного обслуживания цифровых систем на кристаллах // АСУ и приборы автоматики.– Вып. 138. – 2007.– С.83-99.
12. Хаханов В.И., Хаханова И.В. VHDL + Verilog = Синтез за минуты.– Х.: СМІТ, 2007. – 264 с.
13. Zorian Y. What is Infrastructure IP? // IEEE Design & Test of Computers.– May-June 2002. – P. 5-7.
14. Zorian Y., Gizopoulos Dmytris Gest editors' introduction: Design for Yield and reliability // IEEE Design & Test of Computers.– May-June 2004.– P. 177-182.
15. IEEE 1500 Web Site. [Электрон. ресурс]. – Режим доступа: –<http://grouper.ieee.org/groups/1500/>.
16. Densmore D., Passerone R., Sangiovanni-Vincentelli A. A Platform-Based taxonomy for ESL design // Design&Test of computers. – September-October 2006. – P. 359-373.
17. DaSilva F., Zorian Y., Lee Whetsel, Karim Arabi, Rohit Kapur Overview of the IEEE P1500 Standard // ITC International Test Conference.– 2003.– P. 988–997.
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. – 306 с.

*Поступила в редакцию 11.02.2008*

**Рецензент:** д-р техн. наук, проф. В.И. Долгов, Харьковский национальный университет радиоэлектроники.