

# ДЕДУКТИВНЫЙ МЕТОД КУБИЧЕСКОГО МОДЕЛИРОВАНИЯ НЕИСПРАВНОСТЕЙ ЦИФРОВЫХ УСТРОЙСТВ

ХАХАНОВ В.И., ШКИЛЬ А.С., ХАНЬКО В.В.

Предлагается метод дедуктивного моделирования константных логических неисправностей для цифровых устройств (ЦУ), где функциональные элементы представлены в виде кубических покрытий. Впервые получено аналитическое выражение формального анализа покрытия на тест-векторе для транспортирования входных списков неисправностей на выход примитива. Метод инвариантен по отношению к формам представления цифровых схем, задающим способы описания вентилей, функциональных элементов, алгоритмов. Известные дедуктивные формулы анализа вентильных схем являются частным случаем полученного выражения.

## 1. Введение

Усложнение программного и аппаратного обеспечения компьютерных систем увеличивает количество отказов и неисправностей, вызывающих значительные материальные и моральные убытки. Даже аддитивное наращивание в компьютере программных продуктов (аппаратуры) приводит к конфликтным ситуациям, сбоям, отказам, что объясняется отсутствием верификации системной работоспособности и совместимости установленных компонентов. Существующая тенденция ограничения инвестиций на сервисное диагностическое обслуживание компьютерных систем на стадиях проектирования, производства и эксплуатации нуждается в корректировке [1]. Она касается необходимости разработки и сопровождения тестопригодного программно-аппаратного обеспечения компьютера на функциональном, алгоритмическом и системном уровнях. Для этого нужно решать задачи проектирования тестов, алгоритмов контроля и поиска дефектов, системной верификации совместимости компонентов. Решение упомянутых задач требует дальнейшего развития и совершенствования методов и средств моделирования исправного поведения и неисправностей компьютерной структуры на функциональном, алгоритмическом и системном уровнях. Что касается моделирования неисправностей, в настоящее время необходимы новые технологии определения валидности (качества) тестов проверки дефектов компонентов компьютерной системы. Такие технологии должны отвечать требованиям быстродействия, унификации, простоте программной реализации. Этим объясняется актуальность предлагаемой новой технологии моделирования неисправностей ЦУ компьютерных систем.

Общей концепцией модели аппаратного изделия (объекта) можно считать конечный абстрактный автомат первого рода, определяемый аналитическим выражением

$$W = \langle X, Y, Z, f, g, \rangle,$$

где  $X, Y, Z$  – множества входных, выходных и внутренних состояний автомата;  $f, g$  – характерис-

тические функции переходов и выходов, определяющие импликативные отношения в следующем виде:

$$Z(t) = f[X(t), Z(t-1)]; Y(t) = g[X(t), Z(t-1)].$$

Модель – структура компонентов и/или процедур, с определенной степенью адекватности представляющая объект или процесс [2].

Дефект  $F = \{F_p, F_n, F_o\}$  – каждое отдельное несоответствие цифрового изделия требованиям нормативной документации [3, 4]. Повреждение ( $F_p$ ) – вид дефекта, определяемый событием, заключающимся в нарушении исправного состояния при сохранении работоспособного. Несущественный дефект ( $F_n$ ) – событие, которое заключается в нарушении работоспособного состояния при сохранении правильного функционирования. Отказ ( $F_o$ ) – событие, заключающееся в нарушении правильного функционирования. Рейтинг дефектов по деструктивным возможностям определяется соотношением ( $F_p < F_n < F_o$ ).

Техническое состояние объекта – совокупность исправного и множества наперед заданных неисправных состояний:  $S = \{S_i, S_n, = \{S_r, S_f, S_{nf}, S_p\}\}$ . Исправным ( $S_i$ ) называется состояние объекта, при котором он соответствует всем требованиям нормативно-технической документации. Неисправным ( $S_n$ ) называется состояние объекта при наличии в нем дефекта. Работоспособным ( $S_r$ ) называется состояние объекта, при котором он может выполнять свои функции при наличии повреждения. Состояние объекта, при котором он может выполнять свои функции при наличии незначительного дефекта, называется состоянием правильного функционирования ( $S_f$ ). Состояние неправильного функционирования ( $S_{nf}$ ) определяется объектом, имеющим отказ. Если объект имеет катастрофические отказы ( $F_k$ ) или морально устарел на данный момент времени, его техническое состояние определяется как предельное ( $S_p$ ), не подлежащее техническому обслуживанию. Неисправное техническое состояние при наличии дефекта, устранение которого восстановит состояния правильного функционирования, работоспособного или исправного, называется непредельным  $S_{np} = \{S_r, S_f, S_{nf}\}$ .

Тест – вход-выходная последовательность, предназначенная для установления соответствия состояния изделия заданным техническим состояниям:

$$E = || E_{ij}^x, E_{ir}^y ||, i=1, n; j=1, m; r=n+1, k; n – \text{длина теста}; m, k – \text{число входов, линий в объекте}.$$

Дефект  $F_r$  обнаруживается тестом  $E$ , если его присутствие искажает хотя бы одно выходное слово фактической реакции объекта  $R_{ij}^y$  по отношению к его исправному состоянию:

$$\exists ij (E_{ij}^y \cap R_{ij}^y |_{F_r}) = \emptyset. \quad (1)$$

Тип дефекта определяет вид технического состояния [3] изделия, которое может быть: исправным – неисправным, работоспособным – неработоспособным, правильного – неправильного функционирования. Далее не рассматриваются первые две пары состояний, поскольку они определяются дефектами, несущественными для правильного выполнения алгоритма функционирования, называемыми повреждениями. Неисправность часто отождествляется с понятием дефекта.

Методы моделирования неисправностей [2, 5-9], как универсальные средства для решения задач генерации тестов, определения их качества, контроля и поиска дефектов, более сложные в реализации по отношению к алгоритмам анализа исправного поведения, которые являются частными случаями первых. Поэтому решение проблемы моделирования дефектов предполагает существование алгоритмов анализа исправного поведения. Сущность моделирования неисправностей состоит в определении влияния одного или нескольких дефектов на состояние линий объекта при подаче тестовых последовательностей. На сегодня известны десятки типов неисправностей, сгруппированных в классы [5]. Здесь и далее рассматриваются логические одиночные константные неисправности (ОКН), если не оговорено иное. Другими логическими неисправностями могут быть короткие замыкания контактов, их перепутывание, изменения функций элементов.

Большинство практически реализуемых методов использует для анализа неисправностей синхронные модели исправного поведения. На этом основаны алгоритмы одиночного [2], параллельного [5], дедуктивного [5, 9], совместного [6] моделирования.

**Одиночное моделирование неисправностей.** В его основу положена идея внесения одной ОКН эквивалентной линии в схему в целях анализа ее проявления на внешних выходах объекта при подаче тестовых последовательностей. При машинной реализации данный метод требует сравнительно больших затрат времени и ориентирован на обработку любых схем нерегистрового уровня.

**Параллельное моделирование неисправностей.** Используются машинные команды параллельной обработки слов (регистров): логическое сложение, умножение, инверсия, исключающее ИЛИ. Моделирование является компилятивным, поскольку поведение примитивов описывается процедурами на уровне алгоритмических языков или ассемблеров. Одновременно выполняются анализ  $P$  неисправностей на одном входном наборе ( $P$  – разрядность машинного слова, доступного для параллельной обработки). Недостатком метода является ориентация на конкретную ЭВМ, сложность проектирования моделей и проблемных программ. Быстродействие метода в  $P$  раз выше одиночного моделирования.

## 2. Дедуктивное моделирование неисправностей

На входном наборе одновременно обрабатываются все ОКН схемы, выделяя при этом подмножество проверяемых. Анализ примитивного элемента (ПЭ) сводится к выполнению операций над списками неисправностей с помощью аналитических выражений, уникальных для каждого примитива. Сложность получения аналитических преобразователей затрудняет широкое использование метода. Модель элемента (рис. 1) для выполнения дедуктивного алгоритма имеет  $k$  входов и один выход, которым соответствуют значения двоичных сигналов  $Y_1, Y_2, \dots, Y_j, \dots, Y_k, Y_i$ .

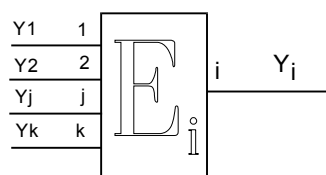


Рис. 1. Модель примитивного элемента

Для каждого входа  $j$  известен входной список неисправностей для фиксированного значения  $Y_j$ . Выходной список  $Z_i$  при заданном  $Y_i$  зависит от функции элемента, значений сигналов на его входах, входных списков неисправностей. Для ПЭ И (И-НЕ) формула получения выходного списка имеет вид [5]

$$\begin{cases} S_i = YS_j \cup \{\equiv \bar{Y}_i\}; \\ \forall j \in J, J_0 = \emptyset \\ S_i = \bigcap_{j \in J_0, J_0 \neq \emptyset} S_j \setminus YS_k \cup \{\equiv \bar{Y}_i\}, \\ \forall k \in J_1 \end{cases} \quad (2)$$

где  $J_0$  ( $J_1$ ) – подмножество входов, имеющих значение 0 (1),  $J = \{J_0, J_1\}$ ;  $\{\equiv \bar{Y}_i\}$  – неисправность выхода  $i$ , инверсная по знаку к его исправному состоянию.

Для элемента ИЛИ (ИЛИ-НЕ) формула содержит операции:

$$\begin{cases} S_i = YS_j \cup \{\equiv \bar{Y}_i\}; \\ \forall j \in J, J_1 = \emptyset \\ S_i = \bigcap_{j \in J_1, J_1 \neq \emptyset} S_j \setminus YS_k \cup \{\equiv \bar{Y}_i\}. \end{cases} \quad (3)$$

Генерация списков неисправностей для внешних входов выполняется в соответствии с выражением  $Z_j = \{\equiv \bar{Y}_j\}$ .

Для реализации дедуктивного алгоритма необходимо получить аналитические формулы для каждого типа ПЭ или преобразовать схему к вентиляльному уровню в базе И-ИЛИ-НЕ, для которого имеются выражения вычисления списков. Рассмотренный метод требует больших затрат памяти для хранения списков, а его эффективность зависит от программной реализации операций над множествами. В упомянутом изложении он ориентирован на вентиляльный уровень описания модели объекта, как и метод параллельного моделирования. Однако аналитические выражения вычисления списков транспортируемых неисправностей можно заменить на явные условия модификации выходов при наличии ОКН. Например, для ПЭ ИИ при входном слове 11 проверяемые дефекты определяются кубами ( $1^0: 010, 2^0: 100, 3^0: 000$ ). Его работа при анализе последовательных схем может иметь значительное число итераций вследствие изменения входного слова или наличия неисправностей, вызывающих генераторный режим в ЦУ. Это приводит к модификации списков проверяемых дефектов на каждой итерации, а значит, к повторению времяемкой процедуры пересечения и объединения подмножеств проверяемых неисправностей. Дедуктивный алгоритм менее предпочтителен, если среднее число вентиляей, входы которых активизированы неисправностями на моделируемом тест-векторе, соизмеримо с общим числом вентиляей.

## 3. Совместное моделирование неисправностей

В совместном (concurrent) методе моделирования [6], как и в дедуктивном, за один проход по схеме определяются все неисправности, проверяемые на тест-векторе. Но в отличие от последнего, где дефекты моделируются дедуктивно (неявно), конкурентный алгоритм анализирует явно исправную работу и те неисправности, которые модифицируют состояния входов или выходов. Класс обрабатываемых дефектов при совместном моделировании включает ОКН, кратные константные неисправности (ККН), а также функциональные. Для примитива модель

неисправности задается кубом, который модифицирует состояние выхода или входа относительно вектора исправного поведения для рассматриваемого ПЭ. Неисправностный список элемента для ОКН есть функция от входного слова. Например, для полусумматора с выходами суммы и переноса при входном векторе 00, вектор моделирования которого равен 0000, такой список имеет вид:  $A^1=1010, B^1=0110, S^1=0010, P^1=0001$ . Внесение неисправности по входу ПЭ требует анализа КП с целью определить состояния его выходов. Для задания дефекта на выходной линии следует записать его тип в соответствующую координату вектора исправного моделирования. При обработке схемы для каждого ПЭ относительно вектора исправного поведения определяется подмножество дефектов, транспортируемых через анализируемый элемент. При этом важно, чтобы общий неисправностный список примитива был упорядочен в соответствии с возрастанием номеров линий схемы (возможен и другой порядок), что исключает переборный характер процедуры анализа условий транспортирования дефекта через ПЭ. Построение входных условий проверки неисправности основано на маскировании состояний входов ПЭ. Если, например, для четырехходового конъюнктора на входном наборе 0110 имеются внешние дефекты ( $7^1, 7^1$ ) на первом и четвертом входах, то куб неисправности  $7^1$  равен 1111, который после моделирования КП доопределяется значением выхода и принимает вид 11111. Видимость кратного дефекта на входах элемента есть следствие наличия в схеме сходящегося разветвления. Чтобы выявить двоичную или большую кратность, следует выполнять сравнение неисправностей линий, которые являются соседними элементами в неисправностном списке ПЭ.

Алгоритм анализа ПЭ при совместном моделировании неисправностей эквипотенциальных линий включает пункты:

1. Исправное моделирование входного набора.
2. Определение кубов входных неисправностей и их моделирование. (Если ОКН входной линии проверяется, то на выходе рассматриваемого ПЭ обнаруживаются все внешние ОКН, протранспортированные к данному входу).

3. Определение кубов кратных внешних дефектов и их моделирование. (Если внешние неисправности одиночные, то условия их транспортирования задаются в п.2).

4. Сравнение состояний выходов ПЭ при отсутствии и наличии дефектов в целях определения выходного списка проверяемых ОКН.

5. Если выходы ПЭ есть внешние линии схемы, то следует во множество обнаруженных неисправностей включить инверсные (по отношению к исправному поведению) значения дефектов выходных линий.

Совместное моделирование можно представить как дедуктивный алгоритм, когда вычисление выходных списков неисправностей является результатом явного моделирования искаженного ими входного слова с последующим сравнением полученных значений при наличии и отсутствии неисправностей. В погоне за уменьшением временных затрат и объема памяти в совместном алгоритме принесена в жертву технологичность анализа объекта диагностирования.

Стоимость совместного алгоритма определяется двумя составляющими – обработкой исправной схе-

мы и неисправностей и зависит от структурной реализации: для линейных схем затраты определяются оценкой  $n^2$ , для квадратной топологии объекта –  $n^{3/2}$ . Таким образом, стоимость выполнения совместного алгоритма зависит в большей степени от топологии схемы (количества линий разветвления).

#### 4. Кубическое моделирование неисправностей

Предлагаемый ниже метод возник на основе разработки алгоритмов и идей, предложенных в [2, 5–9]. Концепция моделирования основана на представлении функционального элемента в формате

$$F^2 = \langle (t-1, t), (X, Z, Y), \{A^2\} \rangle,$$

где  $(t-1, t)$  – два автоматных соседних такта в описании системы функциональных отношений;  $(X, Z, Y)$  – вектор упорядоченных подмножеств входных внутренних выходных переменных;  $\{A^2\}$  – двухтактный алфавит описания состояний автоматных переменных [2]:

$A^2 = \{Q=00, E=01, H=10, J=11, O=\{Q, H\}, I=\{E, J\}, A=\{Q, E\}, B=\{H, J\}, S=\{Q, J\}, P=\{E, H\}, C=\{E, H, J\}, F=\{Q, H, J\}, L=\{Q, E, J\}, V=\{Q, E, H\}, Y=\{Q, E, H, J\}, A^1=\{0, 1\}, X=\{0, 1\}\}, \emptyset(U)\}$ , кубическим покрытием

$$C = (C_1, C_2, \dots, C_i, \dots, C_n),$$

здесь  $C_i = (C_{i1}, C_{i2}, \dots, C_{ij}, \dots, C_{ik})$ , разбитый на упорядоченные подмножества  $C_i = (C_i^X, C_i^Z, C_i^Y)$ . В частном случае, когда функциональный элемент – комбинационный автомат, задаваемый в формате

$$F^1 = \langle (t), (X, Y), \{A^1\} \rangle,$$

кубическое покрытие определяется отношениями на векторе переменных  $C_i = (C_i^X, C_i^Y)$ , что задает многовыходовую комбинационную схему в виде примитивного элемента с  $m$  входами и  $k$  выходами. Если  $k=1$ , речь идет о двоичной булевой функции от  $m$  переменных  $Y=f(X_1, X_2, \dots, X_m)$ .

Кубическое покрытие одновыходовой функции есть совокупность нулевых и единичных кубов – строк таблицы истинности, которые имеют значения нуля, единицы на переменной, идентифицирующей значение функции. Естественно, по единичным кубам можно записать минимизированную дизъюнктивную нормальную форму:

$$f(X_1 \dots X_m) = \bigvee_{\forall Y_i=1} \left( \bigwedge_{j=1}^m X_j^{C_{ij}} \right), \quad (4)$$

по нулевым – обратную дизъюнктивную нормальную форму (ОДНФ):

$$\bar{f}(X_1 \dots X_m) = \bigwedge_{\forall Y_i=0} \left( \bigvee_{j=1}^m \bar{X}_j^{\bar{C}_{ij}} \right). \quad (5)$$

Если переменная  $X_j$  в кубе  $C_i$  принимает значение  $X$ , то она отсутствует в терме, формирующем отношения переменных в упомянутом кубе. При  $C_{ij}=0$  (1) значение  $X_j$  (не) инвертируется для (обратной) дизъюнктивной нормальной формы. Обобщенная функциональная структура кубического покрытия является избыточной и представляется в виде

$$f(X_1 \dots X_m) = \left[ \bigvee_{\forall Y_i=1} \left( \bigwedge_{j=1}^m X_j^{C_{ij}} \right) \right] \vee \left[ \bigwedge_{\forall Y_i=0} \left( \bigvee_{j=1}^m \bar{X}_j^{\bar{C}_{ij}} \right) \right], \quad (6)$$

где каждый куб есть конъюнктивный терм, которому соответствует в общем случае  $m$ -входовой конъюнктор. Все термы, нулевые и единичные, объединяются с помощью дизъюнктора, который имеет число входов, равное количеству кубов  $n$  в покрытии  $S$ .

Определение 1. Список входных неисправностей ПЭ S есть вектор подмножеств проверяемых на входных линиях дефектов  $S = (S_1, S_2, \dots, S_j, \dots, S_m)$ , где  $S_j = \{S_{j1}, S_{j2}, \dots, S_{jr}, \dots, S_{jp_j}\}$ ;  $p_j$  – мощность подмножества проверяемых неисправностей  $S_j$  на входе  $j$ .

Определение 2. Список входных неисправностей ПЭ называется одиночным, если выполняется условие  $S_1 \cap S_2 \cap \dots \cap S_j \cap \dots \cap S_m = \emptyset$ .

Определение 3. Список входных неисправностей ПЭ называется кратным, если выполняется условие  $\exists i, j (S_i \cap S_j) \neq \emptyset$ . Кратность дефекта  $S_{jr}$  определяется максимальным количеством входов, пересечение списков которых равно  $S_{jr}$ .

Определение 4. Булевой производной функции  $f(X_1, \dots, X_i, \dots, X_n)$  называется функция, определяемая выражением [10]:

$$\begin{aligned} \frac{df}{dX_i} &= f(X_1, \dots, X_i, \dots, X_n) \oplus \\ &\oplus f(X_1, \dots, \bar{X}_i, \dots, X_n) = \\ &= f(X_1, \dots, X_{i-1}, 1, X_{i+1}, \dots, X_n) \oplus \\ &\oplus f(X_1, \dots, X_{i-1}, 0, X_{i+1}, \dots, X_n). \end{aligned} \quad (7)$$

Производная по переменной  $X_i$  есть функция, которая не зависит от  $X_i$ . Например, производная

$$\begin{aligned} \frac{d(\bar{X}_2 \bar{X}_3 \vee X_1 X_2 X_3)}{dX_1} &= (\bar{X}_2 \bar{X}_3 \vee X_2 X_3) \oplus (\bar{X}_2 \bar{X}_3) = \\ &= (\bar{X}_2 \bar{X}_3 \vee X_2 X_3) \overline{(\bar{X}_2 \bar{X}_3)} \vee \overline{(\bar{X}_2 \bar{X}_3 \vee X_2 X_3)} (\bar{X}_2 \bar{X}_3) = \\ &= X_2 X_3 \end{aligned} \quad (8)$$

есть неявно задаваемые входные условия, в данном случае  $(X_2 X_3) = (11)$ , при которых активизация входной переменной  $x_1$  приводит к очувствлению выходной.

Определение 5. Теоретико-множественной производной функционала  $f(X_1, \dots, X_i, \dots, X_n)$ , представленного кубическим покрытием в многозначном алфавите, называется функция, определяемая выражением

$$\begin{aligned} \frac{df}{dX_i} &= f(X_1, \dots, X_i, \dots, X_n) \cap \\ &\cap f(X_1, \dots, \bar{X}_i, \dots, X_n) = \\ &= f(X_1, \dots, X_{i-1}, \sigma, X_{i+1}, \dots, X_n) \cap \\ &\cap f(X_1, \dots, X_{i-1}, \bar{\sigma}, X_{i+1}, \dots, X_n), \end{aligned} \quad (9)$$

где  $\sigma, \bar{\sigma}$  – многозначное значение переменной  $X_i$  и ее дополнение. В частном случае, когда значность алфавита равна двум, выражение (9) становится эквивалентным (7).

Определение 6. Производная называется жесткой (H-производная), если она определяет булеву функцию – условия активизации выходной переменной при изменении входной, заданную в одном временном фрейме.

Определение 7. Производная называется мягкой (S-производная), если она определяет булеву функцию – условия активизации выходной переменной при изменении входной, заданную в двух временных фреймах.

Определение 8. Относительно выхода  $r$  множество кубов покрытия  $C$  комбинационного функци-

онала дифференцируется на два непересекающихся подмножества  $C = \{C_i^{r0}, C_i^{r1}\}$ .

**Лемма 1.** Вычисление H-производной комбинационного функционала  $f$  для выхода  $r$  по переменной  $X_j$  определяется пересечением подмножеств нулевых и единичных кубов КП, имеющих на переменной  $X_j$  различные двоичные значения:

$$\frac{df_r}{dX_j} = C_r^H = \{C_{it}^{r0} \mid_{\forall i(q(\exists!j)(C_{ij}^{r0} \cap C_{qj}^{r1} = \emptyset))} \bigotimes_{t=\overline{1,k}} C_{qt}^{r1}\}, \quad (10)$$

где операция пересечения выполняется по правилам:

$$\begin{array}{c|ccc} 1 & 0 & 1 & \bar{0} \\ \hline 0 & 0 & \emptyset & 0 \\ 1 & \emptyset & 1 & 1 \\ \hline \bar{0} & 0 & 1 & \bar{0} \end{array}. \quad (11)$$

Доказательство. Условия вычисления производной по (7) предполагают фиксацию переменных в двух логических значениях 0 и 1. Но поскольку кубическое покрытие есть таблица явных решений, то для вычисления производной следует найти те строки-кубы альтернативных подмножеств  $\{C_i^{r0}, C_i^{r1}\}$ , в которых имеются, во-первых, на координатах  $C_{ij}^{r0} = 0$  (1), а на  $C_{ij}^{r1} = 1$  (0). Во-вторых, непрерывным условием существования производной по  $X_j$  есть наличие пустого пересечения только по входной координате  $j$  упомянутых кубов покрытия. По другим входным линиям КП пересечение не должно быть пустым. Кроме того, пустым должно быть пересечение рассматриваемых кубов по выходной линии  $r$  многовыходного функционала  $f$ , что гарантируется альтернативностью кубов, участвующих в пересечении  $\{C_i^{r0}, C_i^{r1}\}$ . Иначе, принадлежность пе-

ресекаемых кубов к подмножествам  $C_i^{r0}, C_i^{r1}$  гарантированно обеспечивает активизацию функционала  $f_r$  при изменении  $X_j$ . Существование производной

$\frac{df_r}{dX_j} = C_r^H \neq \emptyset$  определяется наличием хотя бы одного куба, вычисленного по правилам (10).

**Следствие 1.** Если  $\frac{df_r}{dX_j} = C_r^H = \emptyset$ , то переменная  $X_j$  является несущественной.

**Лемма 2.** Вычисление S-производной комбинационного функционала  $f$  для выхода  $r$  по переменной  $X_j$  определяется конкатенацией подмножеств нулевых и единичных кубов КП, имеющих на переменной  $X_j$  различные двоичные значения:

$$\frac{df_r}{dX_j} = C_r^S = \{C_{it}^{r0} \bigotimes_{\forall i(q(\exists!j)(C_{ij}^{r0} \cap C_{qj}^{r1} = \emptyset))} \bigotimes_{t=\overline{1,k}} C_{qt}^{r1}\}, \quad (12)$$

где операция конкатенации  $\bigotimes$  выполняется по следующим правилам:

$$\begin{array}{c|ccc} \bigotimes & 0 & 1 & \bar{0} \\ \hline 0 & Q & E & A \\ 1 & H & J & B \\ \hline \bar{0} & O & I & Y \end{array}.$$



Если учесть, что идентификаторы дефектов, как правило, представлены десятичными числами, то неисправности (22) можно записать в виде матрицы или в форме списков:

$$\boxed{\phantom{0000000000}} = \boxed{\phantom{0000000000}}. \quad (23)$$

Смысл дедуктивного метода моделирования неисправностей заключается в вычислении выходного списка проверяемых дефектов за одну итерацию анализа функционала. Следующая теорема обосновывает такую возможность при использовании таблицы истинности в качестве формы представления модели примитива.

**Теорема 1.** Для функционального элемента, заданного таблицей истинности, вычисление всех  $2^m$  производных, формирующих выходной список неисправностей, определяется в соответствии с процедурой следующим выражением:

$$\boxed{\phantom{0000000000}} \quad (24)$$

где  $S_j$  – список входных дефектов на линии  $j$ ;  $E_r, E_j$  – выходная и входная координаты вектора исправного моделирования;  $\boxed{\phantom{0000000000}}$  – неисправность выхода примитива, являющаяся теоретико-множественным дополнением по отношению к его исправному состоянию;  $\boxed{\phantom{0000000000}}$  – входная и выходная координаты  $j$ -го куба покрытия  $S$ ,  $n$  – число кубов;  $k$  – количество невыходных линий;  $m$  – общее число линий в примитиве;  $r=1$  – число выходов.

Доказательство. Основывается на эквивалентности строк таблицы истинности – термов ДНФ и обратной ДНФ, в соответствии с (4) – (6), элементам конъюнкции. Учитывая факт, что оба типа термов равнозначны в смысле формирования функции, в процедуре транспортирования входных списков на выход функционала участвует подмножество строк ДНФ или ОДНФ. Такое разделение обусловлено необходимостью пустого пересечения тест-вектора  $E$  со строками таблицы по выходу функционала в соответствии с (18) – (20), которые формируют списки по  $\boxed{\phantom{0000000000}}$ . Объединение функциональных отношений строк ДНФ или ОДНФ соответствует элементу дизъюнкции. Обработка же элементов конъюнкции и дизъюнкции в совокупности ранее была формализована в виде выражений (2), (3). Таким образом, процедура обработки примитива, представленного таблицей истинности, сводится к объединению результатов формирования выходных списков, вычисленных на основе анализа строк, отождествляемых с логическими элементами конъюнкции. При этом формула получения выходного списка является универсальной, как универсальна сама форма – таблица истинности.

**Теорема 2.** Вычисление всех  $2^m$  производных, формирующих выходной список неисправностей функционала на основе использования выражения (24), инвариантно по отношению к многообразию его кубических покрытий.

Доказательство. Кубическое покрытие есть результат минимизации таблицы истинности известными методами, которые дают множество тупиковых минимизированных форм, вследствие неоднозначности решения задачи минимизации. Формально кубическое покрытие отличается от таблицы истинности наличием символов  $X$  на координатах кубов. Рассмотрим влияние этих символов на получение списка проверяемых на выходе функционала дефектов в сравнении с двоичной таблицей истинности. Для двух двоичных кубов  $S_i, S_t$ , подлежащих склеиванию по только одной входной координате  $j$ :  $\boxed{\phantom{0000000000}}$  и удовлетворяющих условию (24), можно записать, что

$$\boxed{\phantom{0000000000}} \quad (25)$$

поскольку для двух кубов  $S_i, S_t$ , отличающихся по  $j$ , список  $S$ , полученный по координатам, отличным от  $j$ , взаимодействует с  $S_j$  на первом кубе как  $\boxed{\phantom{0000000000}}$ , на втором –  $\boxed{\phantom{0000000000}}$ . Если  $\boxed{\phantom{0000000000}}$  то в этом случае получается  $\boxed{\phantom{0000000000}}$ , что доказывает истинность выражения (25). Иначе,  $\boxed{\phantom{0000000000}}$ , что также доказывает тождественность (25). И последний случай, когда  $\boxed{\phantom{0000000000}}$ , выражение (25) превращается в  $\boxed{\phantom{0000000000}}$ . Таким образом, выходной список  $S$ , формируемый двумя координатами строк таблицы истинности, инвариантен списку, полученному по одной строке покрытия, которая на  $S_{ij}$  имеет символ  $X$ . Наличие же нескольких символов  $X$  в кубе  $S_{ij}$  всегда сводится к ситуации (25). Следовательно, результат – выходной список неисправностей  $S$ , полученный на основе обработки таблицы истинности по (24), эквивалентен списку, вычисленному на основе анализа любой тупиковой формы кубического покрытия функционала.

**Следствие 2.** Универсальная формула определения списка неисправностей  $S_r$  по выходу  $r$  кубического покрытия  $S$  (таблицы истинности) многовыходового примитивного элемента имеет вид

$$\boxed{\phantom{0000000000}} \quad (26)$$

$n$  – число кубов;  $k$  – количество невыходных линий;  $m$  – общее число линий в примитиве.

**Пример 1.** Определение выходного списка неисправностей на тест-векторе 1111 для функционала, заданного кубическим покрытием (13), где входные списки определены подмножествами:

$$\boxed{\phantom{0000000000}}$$

сводится к выполнению пересечений сначала по выходным координатам вектора E и кубов покрытия, которые формируют кубы  $C_i$ , удовлетворяющие условиям необходимости [redacted]. Далее анализ упомянутых кубов по (26) дает результат в данном случае обработки каждого куба покрытия:

$$[redacted]$$

где  $S^i$  – частичный список, формируемый соответствующим кубом  $C_i$ ;  $S(E)$  – обобщенный список дефектов, проверяемый тест-вектором E и получае-

мый объединением частичных списков: [redacted].

Для тест-вектора 0100 результат моделирования тех же неисправностей на входах имеет вид:

$$[redacted]$$

**Теорема 3.** Процедура определения выходного списка дефектов S элемента (схемы) на тест-векторе E, имеющем n символов X и двоичное состояние выхода, задается выражением

$$[redacted] \quad (27)$$

где  $S^t$  – выходной список, проверяемый двоичным тест-вектором [redacted], где мощность множества двоичных векторов равна [redacted].

Доказательство. Поскольку вектор E существует в одном временном фрейме, то проверяющие способности каждого из двоичных наборов [redacted] не должны противоречить друг другу, что возможно лишь в случае, когда пересечение выходных списков не равно пустому множеству: [redacted]. Возможен и противный случай, когда [redacted]. Такой результат имеет место, когда все проверяемые на двоичных составляющих наборах [redacted] неисправности относятся к переменной j. Тогда по аналогии с (25) тождество

$$[redacted] \quad (28)$$

дает основание утверждать, что список проверяемых неисправностей для тест-вектора, имеющего символ X, будет пуст. То же можно сказать и для наборов, содержащих n символов X, поскольку такие векторы можно разложить на составляющие, содержащие по одному символу X.

**Следствие 3.** Если состояние выхода функционала на троичном тест-векторе [redacted] равно X, то выходной список неисправностей равен [redacted].

Здесь по выходным координатам не выполняется условие пустого пересечения, определенное в (24). Отсюда следует доказательство справедливости предлагаемого следствия.

На основе процедуры обработки примитива, заданного кубическим покрытием, определим алгоритм дедуктивного моделирования неисправностей схемы цифрового устройства, реализующего конечный автомат:

1. Моделирование исправного поведения очередного функционального элемента [redacted] на тест-векторе [redacted]. Если [redacted] – формирование списка S(E) проверенных неисправностей на тесте E – конец моделирования. Иначе, [redacted] – п.2.

2. Если все элементы схемы обработаны [redacted], выполняется сравнение исправных состояний линий в двух соседних тактах. Если состояния идентичны [redacted], конец моделирования очередного тест-вектора [redacted] и переход к п. 3. Иначе – п. 1.

3. Генерирование списков неисправностей внешних входов схемы, как теоретико-множественного дополнения их исправного состояния [redacted].

4. Моделирование неисправностей функционального элемента [redacted] по процедуре (26).

5. Если [redacted], выполняется формирование списка дефектов  $S(E^t)$ , проверяемых на тест-векторе  $E^t$ , и переход к п.1. Иначе, [redacted] – п. 4.

Моделирование неисправностей рассмотрим на примере анализа схемы, представленной на рис.2, где поведение функциональных элементов задается кубическими покрытиями:

$$[redacted]; [redacted]; [redacted].$$

Рассматриваются одиночные константные неисправности эквипотенциальных линий, обозначаемые в виде [redacted], где [redacted] – знак константной неисправности, j – номер линии схемы.

Для каждого входного набора, в данном случае E=1111, моделируется исправное поведение объекта, которое дает результат по всем линиям, представленный вектором (111101011).

Априорно генерируется список входных дефектов схемы цифрового устройства:

$$[redacted]$$

Выполняется последовательная обработка всех элементов схемы для вычисления выходных списков неисправностей с учетом (26):

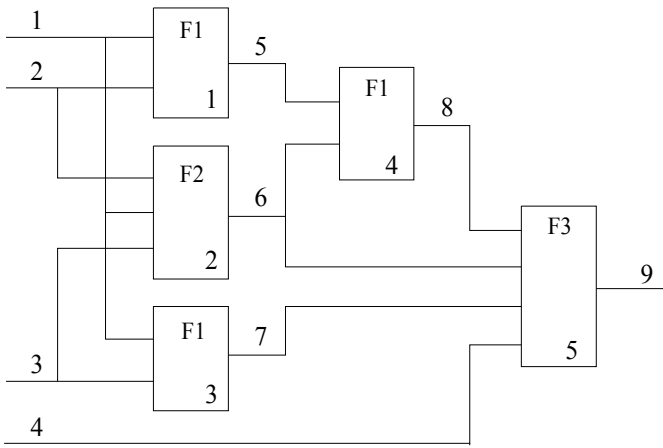


Рис. 2. Цифровая схема для моделирования

$$\begin{aligned}
 S^1(C_1 = 1X0) &= \emptyset; \\
 S^2(C_2 = 011) &= S_1 \setminus S_2 = \{1^0\}; \\
 S^3(C_3 = 000) &= S_1 \cap S_2 = \emptyset; \\
 S_5(E=110) &= S^2 \vee S^3 \vee S^1 = \{1^0, 5^1\}. \\
 \\
 S^1(C_1 = X001) &= \emptyset; \\
 S^2(C_2 = 1111) &= \emptyset; \\
 S^3(C_3 = X010) &= S_1 \setminus S_3 = \{1^0\}; \\
 S^4(C_4 = X100) &= S_3 \setminus S_1 = \{3^0\}; \\
 S^5(C_5 = 01X0) &= S_2 \setminus S_1 = \{2^0\}; \\
 S_6(E=1111) &= S^3 \vee S^4 \vee S^5 \vee S^6 = \{1^0, 2^0, 3^0, 6^0\}. \\
 \\
 S^1(C_1 = 1X0) &= \emptyset; \\
 S^2(C_2 = 011) &= S_1 \setminus S_3 = \{1^0\}; \\
 S^3(C_3 = 000) &= S_1 \cap S_3 = \emptyset; \\
 S_7(E=110) &= S^2 \vee S^3 \vee S^1 = \{1^0, 7^1\}. \\
 \\
 S^1(C_1 = 1X0) &= S_5 = \{1^0, 5^1\}; \\
 S^2(C_2 = 011) &= \emptyset; \\
 S^3(C_3 = 000) &= S_6 \setminus S_5 = \{2^0, 3^0, 6^0\}; \\
 S_8(E=011) &= S^1 \vee S^3 \vee S^8 = \{1^0, 2^0, 3^0, 5^1, 6^0, 8^0\}. \\
 \\
 S^1(C_1 = X00X1) &= \emptyset; \\
 S^2(C_2 = X1011) &= \emptyset; \\
 S^3(C_3 = 011X1) &= \emptyset; \\
 S^4(C_4 = X01X0) &= S_7 \cap S_6 = \{1^0\}; \\
 S^5(C_5 = 1X1X0) &= S_7 \setminus S_8 = \{7^1\}; \\
 S^6(C_6 = X1000) &= S_4 \setminus S_6 \setminus S_7 = \{4^0\}; \\
 S_9(E = 11011) &= S^4 \vee S^5 \vee S^6 \vee S^9 = \{1^0, 4^0, 7^1, 9^0\}.
 \end{aligned}$$

Качество тест-вектора 1111, определяемое соотношением  $Q = [q / (2 \times m)] \times 100\%$  ( $q$  – число проверяемых дефектов,  $m$  – количество линий схемы), равно:  $Q = [4 / (2 \times 9)] \times 100\% = 22\%$ . Так же определяется и качество теста – отношение числа всех проверенных дефектов к удвоенному числу линий.

## 5. Заключение

Предложенный дедуктивный метод моделирования неисправностей является новой, не имеющей

аналогов, технологией обработки цифровых схем функционального уровня представления на основе использования кубических покрытий. Он позволяет за одну итерацию моделировать все неисправности цифрового устройства, проверяемые тест-вектором. Его основное достоинство заключается в инвариантности по отношению к формам представления цифровых схем, задаваемых в виде вентилей, функциональных элементов, алгоритмов, других способов описания, сводимых к таблицам – кубическим покрытиям. Предложенный метод также эффективно обрабатывает и последовательностные функционалы, описанные двухтактными кубическими покрытиями, которые формализуют алгоритмический уровень представления цифровых устройств. Проблема дедуктивного моделирования микропроцессорных структур, описанных в виде языков высокоуровня VHDL, остается открытой.

**Литература.** 1. Courtois B. CAD and testing of ICs and systems. Where are we going? – TIMA, France. 1995. 250 p. 2. Хаханов В.И. Техническая диагностика элементов и узлов персональных компьютеров. К.: ИЗМН. 1997. 308 с. 3. Хаханов В.И., Фрадков С.А., Ханько В.В. Модель неисправностей программного продукта. Компьютерный вирус. Радиоэлектроника и информатика. 1998. №1. С. 98–105. 4. ГОСТ 20911–89. Техническая диагностика. Термины и определения. М.: Изд-во стандартов, 1989. 16с. 5. Автоматизированное проектирование цифровых устройств / С.С.Вадулин, Ю.М.Барнаулов и др. / Под ред. С.С.Вадулина. М.: Радио и связь, 1981. 240с. 6. Abramovici M., Breuer M.A., Kumar K. Concurrent Fault Simulation and Functional Level Modeling // 14-th Design Automata Conf., Milano. 1977. P.128–137. 7. Хаханов В.И., Шкиль А.С. Троичное моделирование цифровых устройств с использованием языка кубических комплексов // АСУ и приборы автоматики. 1983. Вып. 66. С.59–65. 8. Hayes J.P. A systematic approach to multivalued digital simulation // ICCD-84: Proc. IEEE Int. Conf. Comput. 1984. No. 4. P. 177–182. 9. Биргер А.Г. Многозначное дедуктивное моделирование цифровых устройств // Автоматика и вычислительная техника. 1982. №4. С.77–82. 10. Богомолов А.М., Сперанский Д.В. Аналитические методы в задачах контроля и анализа дискретных устройств. Саратов: Изд-во Саратов. ун-та, 1986. 240с.

Поступила в редколлегию 12.03.99

**Рецензент:** д-р техн. наук Кривуля Г.Ф.

**Хаханов Владимир Иванович**, д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 310726, Харьков, пр. Ленина, 14, тел. 40–93–26.

**Шкиль Александр Сергеевич**, канд. техн. наук, доцент кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств. Увлечения: теннис. Адрес: Украина, 310726, Харьков, пр. Ленина, 14, тел. 40–93–26.

**Ханько Вадим Викторович**, аспирант кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика компьютерных систем и сетей. Увлечения: иностранные языки. Адрес: Украина, 310726, Харьков, пр. Ленина, 14, тел. 40–93–26.