

ГЕНЕРАЦИЯ ТЕСТОВ ДЛЯ ПОСЛЕДОВАТЕЛЬНОСТНЫХ СХЕМ С ИСПОЛЬЗОВАНИЕМ КРАТНОЙ СТРАТЕГИИ НАБЛЮДЕНИЯ ВЫХОДНЫХ СИГНАЛОВ

Разработан метод построения текстовых последовательностей для цифровых схем с памятью на основе распознавания пар состояний исправной и неисправной схемы. Используется кратная стратегия наблюдения выходных сигналов, 16-значный алфавит и генетические алгоритмы. Предложенный метод позволяет выявить неисправности, которые не проверяются традиционными методами и тем самым повысить полноту контролирующих тестов.

The test generation method is designed for digital circuits with memory on the basis of distinguishing state pairs of good and fault devices. The multiple observation time test strategy, 16-valued alphabet and genetic algorithms are used. The proposed method permits to cover the faults that are not detected with traditional methods. It increases the fault coverage.

Введение

В диагностике последовательностных цифровых схем в последнее время достигнуты значительные успехи, которые, прежде всего основаны на использовании методов искусственного интеллекта. Но сама постановка задачи построения проверяющих тестов для последовательностных схем существенно зависит от применяемой стратегии наблюдения выходных сигналов. Известно, что стандартные методы моделирования неисправных цифровых схем, основанные на логическом моделировании в троичном алфавите с применением одиночной (обычной) стратегии наблюдения выходных сигналов, не позволяют точно оценить полноту проверяющих тестов [1,2]. Несмотря на многочисленные попытки создать автоматические системы генерации проверяющих тестов для последовательностных цифровых схем, эта проблема далека от окончательного решения вследствие необходимости в общем случае учитывать возможность нахождения в произвольном начальном состоянии как исправной, так и неисправной схемы.

Постановка задачи построения тестовой последовательности для схем с памятью

Пусть исправная последовательностная схема реализует конечный автомат $A=(Y,X,Z,\delta,\lambda)$,

где Y, X, Z - конечные множества состояний, входных и выходных сигналов соответственно; $\delta:Y \times X \rightarrow Y$ - функция переходов, определяющая следующее состояние автомата; $\lambda:Y \times X \rightarrow Z$ - функция выхода, определяющая выходной сигнал. Функции δ и λ реализуются комбинационными схемами, где :

$$Y=(y_1, \dots, y_k) \text{ где } y_i=(0,1) \text{ для } i=\overline{1, k};$$

$$X=(x_1, \dots, x_n) \text{ где } x_l=(0,1) \text{ для } l=\overline{1, n};$$

$$Z=(z_1, \dots, z_m) \text{ где } z_j=(0,1) \text{ для } j=\overline{1, m};$$

Пусть $X(1), X(2), \dots, X(p)$ - обозначает входную последовательность длины p . Тогда $Y(y_0, 0), Y(y_0, 1), \dots, Y(y_0, p)$ - означает последовательность состояний автомата, которую он проходит из начального состояния $y_0 \in Y$ под воздействием входной последовательности $X(1), X(2), \dots, X(p)$. Пусть $Z(y_0, 0), Z(y_0, 1), \dots, Z(y_0, p)$ - обозначает выходную последовательность, производимую автоматом из начального состояния y_0 при подаче входной последовательности $X(1), X(2), \dots, X(p)$. Обозначим через $z_j(y_0, t)$ для $j=\overline{1, m}$ значение j -го выхода на t -м шаге моделирования. Используя эти обозначения, следующее состояние определяется сле-

дующим образом:

$$Y(y_0, t) = \begin{cases} y_0 & \text{для } t = 0 \\ \delta(X(t), Y(y_0, t-1)) & \text{для } t \neq 0 \end{cases}$$

Аналогично выход $Z(y_0, t)$ определяется функцией λ . Неисправность f преобразует автомат A в $A^f = (Y, X, Z, \delta^f, \lambda^f)$, где функции δ^f, λ^f состояния Y^f и выхода Z^f определяются таким же образом.

В качестве примера, рассмотрим функциональные модели исправной схемы, представленной на рисунке 1 и ее неисправной модификации с одиночной константной неисправностью $f_1 \equiv 0$.

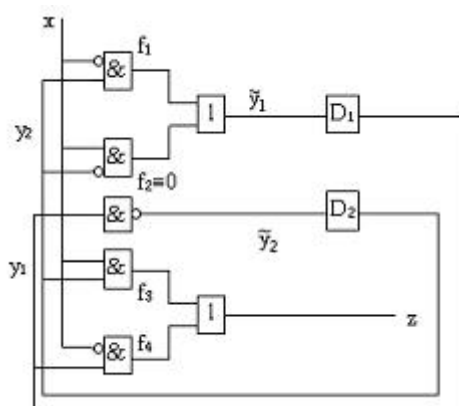


Рис. 1. Пример схемы.

Таблица 1. Исправный автомат

S	$S_{сл, z}$	
	$x=0$	$x=1$
A	B,0	C,0
B	C,0	B,0
C	D,1	A,1
D	A,1	D,0

Таблица 2. Неисправный автомат

$S(y_1 y_2)$	$S_{сл, z}$	
	$x=0$	$x=1$
a	b,0	c,0
b	b,0	b,0
c	a,1	a,1
d	a,1	d,0

Например, в табл.1 и табл.2 приведены автоматы, реализуемые схемой рис.1, исправной и содержащей неисправность $f_1 \equiv 0$ соответственно.

На данный момент существует ряд методов генерации тестов и моделирования с неисправностями последовательных цифровых устройств (ЦУ) [1,2]. Как правило, в них используется стратегия одиночного наблюдения выходных реакций, которую можно определить следующим образом.

Определение 1. Неисправность f называется обнаружимой в последовательной схеме входной последовательностью $X(1), X(2), \dots, X(p)$ относительно стратегии одиночного наблюдения выходов (SOT), если

$$\exists t \leq p, \exists j \leq m, \exists b \in \{0,1\}, \text{ такое, что}$$

$$\forall (r, q) : (z_j(r, t) = b \wedge z_j^f(q, t) = \bar{b})$$

где r – начальное состояние исправной схемы и q – начальное состояние неисправной схемы. Это определение говорит, что при данной стратегии неисправность считается обнаружимой, если найдется (по крайней мере один) момент времени (такт) t такой, что для любой пары состояний (r, q) исправной и неисправной схем некоторый j -й выход z_j имеет различные значения в исправной и неисправной схеме. Ключевым моментом является то, что любая пара состояний исправной и неисправной схемы должна выдать различные выходные реакции в один и тот же такт времени. В табл.3 и табл.4 приведены выходные реакции исправной и неисправной схем соответственно на входную последовательность $T=(0,0,0,0)$, проверяющую данную неисправность. Из таблиц видно, что не существует одного момента времени, когда для всех пар начальных состояний рассматриваемой схемы выходные реакции исправного и неисправного устройства различались бы. То есть, в соответствии со стратегией одиночного наблюдения, данная входная последовательность не является тестом для неисправности $f_1 \equiv 0$, что, как показано далее, неверно. Рассмотрим более точное определение проверяемости для схем с памятью.

Определение 2. Неисправность f называется проверяемой в последовательной схеме входной последовательностью $X(1), X(2), \dots, X(p)$ относительно стратегии кратного наблюдения выходов (MOT), если:

$\forall (r, q) \exists t \leq p, \exists j \leq m, \exists b \in \{0,1\}$ такое, что
 $:(z_j(r, t) = b \wedge z_j^f(q, t) = \bar{b})$

Термин «кратная стратегия наблюдения выходных сигналов» формально был введен в работе [3], хотя, фактически, этот подход использовался одним из авторов задолго до этой публикации, например в [4].

Принципиальное отличие между этими стратегиями состоит в следующем. Согласно первой стратегии неисправность обнаружима (и, следовательно, для нее можно построить тестовую последовательность) если найдется один момент времени t , такой что независимо от начального состояния исправной и неисправной схем значения выходных сигналов различны для исправной и неисправной схем. То есть все пары состояний исправной и неисправной схемы выдают различные выходные сигналы в один и тот же момент времени. Согласно второй стратегии для каждой пары состояний исправной и неисправной схемы может существовать свой момент времени, в котором они дают различные выходные сигналы.

Для приведенного выше примера неисправность $f_1 \equiv 0$ является обнаружимой входной последовательностью $T=(0,0,0,0)$, относительно кратной стратегии наблюдения поскольку для каждой пары начальных состояний исправного и неисправного ЦУ существует момент времени, когда реакции исправной схемы отличаются от реакции неисправной.

Таблица 3. Реакции исправной схемы

S(y ₁ y ₂)	x ₁ =0	x ₂ =0	x ₃ =0	x ₄ =0
A	0	0	1	1
B	0	1	1	0
C	1	1	0	0
D	1	0	0	1

Таблица 4. Реакции неисправной схемы

S(y ₁ y ₂)	x ₁ =1	x ₂ =1	x ₃ =1	x ₄ =1
a	0	0	0	0
b	0	0	0	0
c	1	0	0	0
d	1	0	0	0

Естественно, определение проверяемости неисправности существенно влияет на постановку задачи построения тестовой проверяющей

последовательности. При построении теста для схем с памятью необходимо найти входную последовательность X(1), X(2),..., X(p), для которой выполняется соответствующее определение проверяемости неисправности.

В настоящее время существуют два основных подхода к решению этой проблемы. При первом подходе [3] каждая пара состояний исправной и неисправной схемы (что необходимо при кратной стратегии наблюдения выходных сигналов) обрабатывается индивидуально и полученные результаты «склеиваются». Второй подход [5,6] основан на использовании символического моделирования исправных и неисправных последовательностных схем.

Структурная модель

Напомним, что согласно приведенному определению 2 проверяемости неисправности необходимо различить каждую пару состояний исправной и неисправной схем. Поэтому за основу можно взять входные последовательности, которые различают отдельные пары состояний исправной и неисправной схем.

Наиболее распространенный на практике подход к построению тестов для последовательностных схем основан на преобразовании их в так называемый комбинационный эквивалент (КЭ) [1]. КЭ получается в результате условного обрыва обратных связей схемы как показано на рис.2. Далее строится итеративная комбинационная схема путем последовательностного соединения (КЭ). В результате такого преобразования линии начального состояния последовательностной схемы Y становятся псевдовходами, линии конечного состояния \tilde{Y} - псевдовыходами, что представлено рис.3.

Отметим, что итеративная комбинационная схема в общем случае неадекватно отражает поведение асинхронных последовательностных схем. Строго говоря, эта модель соответствует синхронной последовательностной схеме с синхронизацией в точках обрыва обратных связей.

Эта неадекватность «провоцирует» различную трактовку проверяемости неисправностей в последовательностных схемах, что порождает массу проблем, преодоление которых имеет много нюансов. Различные методы построения тестов решают эти проблемы по-разному. При этом фактически разные исследователи используют различные определения проверяемости

неисправностей для схем с памятью [6], что создает путаницу и затрудняет исследования в данной области. Например, стратегия SOT согласно определению 1 фактически учитывает выходные реакции только последнего КЭ (такта) в этой итеративной схеме. Наоборот стратегия MOT согласно определению 2 учитывает выходные реакции всех КЭ (всех тактов).

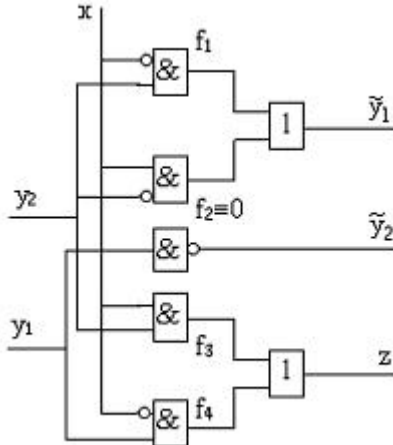


Рис.2. Комбинационный эквивалент последовательностной схемы.

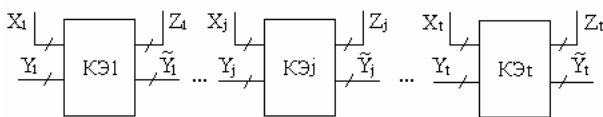


Рис.3. Комбинационная итеративная схема из t КЭ.

Применение стратегии кратного наблюдения позволяет повысить полноту проверяющих тестов, но требует значительных вычислительных ресурсов и затрат памяти. Так, например, при этой стратегии необходимо хранить эталонные выходные реакции исправной схемы для каждого возможного начального состояния. Для неисправной схемы также необходимо вычислить выходные реакции для каждого возможного состояния и сравнить их с эталонными реакциями исправной схемы.

Далее для построения проверяющих тестов мы будем использовать структурную модель в виде итеративной комбинационной схемы, кратную стратегию MOT и многозначный алфавит $V_{16} = \{\emptyset, 1, D, G1, D', F1, D^*, D1, 0, C, F0, H, G0, E, D0, u\}$ [1], где символы алфавита представляют различные ситуации в исправной и неисправной схемах, возможные в процессе генерации тестовой последовательности. При этом анализ двух схем (исправной и неисправной) в двоичном (или троичном) алфавите заменяется

анализом одной схемы в многозначном алфавите [1].

Построение тестовой последовательности на основе различения пар состояний

Итак при генерации тестовой последовательности необходимо различить каждую пару состояний исправной и неисправной схем. Мы будем использовать общий подход [3], который основан на индивидуальном различении каждой пары состояний. В начале процесса построения теста имеется множество пар начальных состояний SI, для которых необходимо построить различающую последовательность T. Очевидно, что в начале процесса генерации теста множество SI эквивалентно множеству всех пар состояний исправного и неисправного устройств. Например, для схемы рис.1 имеем множество $SI = \{(A,a), (A,b), (A,c), (A,d), (B,a), (B,b), (B,c), (B,d), (C,a), (C,b), (C,c), (C,d), (D,a), (D,b), (D,c), (D,d)\}$. Суть метода построения проверяющей тестовой последовательности заключается в следующем. Из множества пар начальных состояний SI берем некоторую пару начальных состояний S^i исправного и S^j_f неисправного устройств: (S^i, S^j_f) . Для этой пары состояний строится различающая последовательность T_1 . Последовательность T_1 генерируется при помощи генетического алгоритма, использующего универсальный 16-значный алфавит и многозначные функции. Однако, данный метод применяется при известных значениях переменных начального состояния, так как пара начальных состояний известна. Поэтому, благодаря отсутствию неопределенного значения и на линиях начального состояния, различающая последовательность генерируется довольно просто. На данном этапе мы полагаем искомую входную последовательность $T = T_1$. Далее проводим моделирование с неисправностью данного устройства на последовательности T в многозначном алфавите V_{16} . Если для последовательности T выполняется критерий проверяемости неисправности согласно определению 2, то полученная входная последовательность T является тестом для рассматриваемой неисправности и процедура генерации теста завершена. Иначе T процесс построения проверяющей последовательности продолжается. Далее определяется новое множество пар начальных состояний SI, для которых необходимо построить различающую

щие последовательности. Новое множество SI состоит из пар состояний, принадлежащих множеству SF за исключением тех, которые принадлежат множеству SD уже различных пар состояний: $SI = SF \setminus SD$.

Из полученного множества SI выбираем некоторую пару начальных состояний (S^2, S_f^2) и строим для нее различающую последовательность T_2 , используя генетический алгоритм. Теперь полагаем искомую входную последовательность T равной конкатенации T и T_2 . Для нее выполняется логическое моделирование с заданной неисправностью в многозначном алфавите. Если для T выполняется критерий проверяемости, то построенная входная последовательность является тестом данной неисправности и процесс генерации завершается, иначе - продолжается. И так далее пока либо не будет найдена тестовая последовательность T , либо количество итераций не превысит установленную границу.

Рассмотрим этот подход на примере схемы представленной на рис. 1. Если исправная и неисправная схемы стартуют из полностью неопределенных состояний, что в многозначном алфавите представляется значениями $y_1 = u, y_2 = u$, то тестовая последовательность не может быть построена с использованием стратегии одиночного наблюдения сигналов (SOT - определение 1), поскольку даже исправная схема дает неопределенные значения выходных сигналов (нет однозначной выходной реакции). Далее попробуем частично уменьшить неопределенность начального состояния исправной схемы, положив значение переменной $y_1 = 0$ и оставив полностью неопределенным состояние неисправной схемы. Для анализа используем универсальный 16-значный алфавит V_{16} , который одновременно позволяет выполнять моделирование исправной и неисправной схем в многозначном алфавите V_{16} . Рассмотрим начальное состояние $y_1 = G0, y_2 = u$, что соответствует следующим значениям сигналов: в исправной схеме переменная состояния $y_1 = 0$, а в неисправной схеме эта переменная имеет неопределенное значение: переменная $y_2 = u$ как в исправной, так и в неисправной схеме. Выполним логическое моделирование (исправной и неисправной схем) на входной последовательности $T=(0,0,0)$

в универсальном алфавите V_{16} . В результате получаем следующие значения выхода $z=(G0,F0,D)$, которые показывают, что в этом случае (при фиксации состояния исправной схемы $y_1 = 0$) исправная и неисправные схемы выдают различные выходные реакции на входную последовательность $T=(0,0,0)$ на третьем такте, что показывает значение $z=D$ (которое означает 1 – в исправной и 0 – в неисправной схемах). Таким образом, мы показали, что два состояния исправной схемы $(y_1, y_2 = (00,01))$ с входной последовательностью $x=(0,0,0)$ отличаются от любых четырех состояний неисправной схемы $(y_1^f, y_2^f = (00,01,10,11))$. Аналогично, путем присваивания начального состояния $y_1 = G1, y_2 = u$ и логического моделирования в V_{16} можно показать, что эта входная последовательность различает оставшиеся пары состояний исправной $(y_1, y_2 = (10,110))$ и неисправной $y_1^f, y_2^f = (00,01,10,11)$ схем. Таким образом, анализ 16 пар состояний исправной и неисправной схем в двоичном алфавите мы заменили анализом двух схем в многозначном алфавите. Следует отметить, что использование универсального 16-значного алфавита и одной схемы по сравнению с анализом по отдельности двух (исправной и неисправной) схем в троичном алфавите [3] позволяет существенно уменьшить сложность проверки (на практике проверка часто выполняется с линейной сложностью). Приведенный пример показывает, что пары начальных состояний исправной и неисправной схем, которые используются при генерации тестовой последовательности на шаге могут быть частично неопределенными. При этом мы шли «сверху вниз», взяв сначала полностью неопределенные значения для состояний и убедившись, что в этом случае нельзя построить тестовую последовательность, начали «снимать» неопределенность для отдельных переменных состояний.

Заметим, что можно действовать наоборот «снизу вверх», взяв за основу одну пару состояний с определенными значениями, и далее вносить неопределенность для отдельных переменных состояний. Предположим, что на текущий момент для некоторой неисправности f построена тестовая последовательность T , которая различает пару состояний (S,Q) исправ-

ной и неисправной схем. Предположим для определенности, что эти состояния определяются следующими значениями переменных $(\delta_1, \delta_2, \dots, \delta_k)$, где каждая δ_i может принимать значения $\{0, 1, D, D'\}$ 16-значного алфавита V_{16} . При выборе новой пары состояний (S', Q') мы попытаемся внести неопределенности в значения $(\delta_1, \delta_2, \dots, \delta_k)$ переменных состояния в как можно большем числе переменных, сохраняя при этом свойство входной последовательности проверять рассматриваемую неисправность для пары состояний (S, Q) (точнее множества пар, включающего (S, Q)). Таким образом, мы расширяем подмножество пар состояний исправной и неисправной схем, для которого входная последовательность остается проверяющей. При этом переменные состояния (y_1, y_2, \dots, y_k) рассматриваются последовательно путем замены определенных значений $\{0, 1, D, D'\}$ алфавита V_{16} более неопределенными значениями следующим образом:

- 1) $y_i=0$ (0 в исправной и 0 в неисправной схемах) $\rightarrow y_i=G0$ или $y_i=F0$;
- 2) $y_i=1$ (1 в исправной и 1 в неисправной схемах) $\rightarrow y_i=G1$ или $y_i=F1$;
- 3) $y_i=D$ (1 в исправной и 0 в неисправной схемах) $\rightarrow y_i=F0$ или $y_i=G1$;
- 4) $y_i=D'$ (0 в исправной и 1 в неисправной схемах) $\rightarrow y_i=F1$ или $y_i=G0$.

Отметим, что при этом для каждой пары состояний должен остаться момент времени, где выходные реакции исправной и неисправной схем отличаются. Если при таком введении неопределенности свойство быть тестовой последовательностью теряется, то для текущей переменной состояния восстанавливается исходное (определенное) значение. Пара состояний (P/R) , в которую перешли исправная и неисправная схемы, может рассматриваться в качестве исходной на следующем этапе построения тестовой последовательности только в том случае, если эта пара не покрывается полученными на предыдущих шагах значениями переменных состояний с неопределенностями.

Учитывая все вышесказанное, в укрупненном виде алгоритм построения тестовой последовательности можно сформулировать следующим образом. Пусть T – формируемая входная тестовая последовательность, а SI – множество пар состояний исправной и неисправной схем, различаемых текущей T .

Генерация тестовой последовательности (схема)

```

{
  T=∅, SI=∅;
  While (есть неразличимые пары состояний)
  {
    Выбор пары (возможно не полностью определенных) состояний (S,Q)
    (S = (α1, α2, ..., αk), Q = (β1, β2, ..., βk)), не
    покрываемых на текущий момент SI;
    If (неразличимой пары не существует)
    Then тест T построен: return;
    Формирование значений переменных
    состояний (δ1, δ2, ..., δk) в алфавите V16;
    логическое моделирование в алфавите
    V16 на входной последовательности T с
    начальными состояниями (δ1, δ2, ..., δk);
    if (значение D или D' достигает
    внешнего выхода схемы)
    then
      пара состояний (S,Q) различается
      текущей последовательностью T;
    Else
    {
      построение различающей последо-
      вательности T (S,Q);
      if (различающая последователь-
      ность для (S,Q) не построена)
      then
        текущая неисправность непрове-
        ряемая и удаляется из дальнейшего
        рассмотрения;
        переход на конец алгоритма;
      }
      Для пары состояний
      S = (α1, α2, ..., αk), Q = (β1, β2, ..., βk)
      For j=1 to k do
        δj = Fβj;
      логическое моделирование в V16;
      if (T не различает (S,Q))
      then
        восстановление прежнего
        значения δj;
        δj = Gαj;
      логическое моделирование в V16;
      if (T не различает (S,Q))
      then
        восстановление прежнего
        значения δj;
    }
  }

```

end
определение новых различаемых пар состояний (\bar{S}, \bar{Q}) ;
определение всех различаемых пар состояний $SI = SI \cup \{\bar{S}, \bar{Q}\}$;
}

Приведенный алгоритм гарантирует построение тестовой последовательности для избыточной неисправности в том случае, если гарантируется построение различающей входной последовательности T для пары (S, Q) .

Задача построения входной последовательности, проверяющей пару состояний (S/Q) исправной и неисправной схемы может быть решена следующим образом. Прежде всего, в итеративной комбинационной схеме псевдоходам первого комбинационного эквивалента, соответствующим переменным состояниям, присваиваются начальные значения, которые определяются парой состояний (S/Q) . Если i -я переменная в исправной схеме $y_i=1$, а в неисправной $-y_i=0$, то этой переменной присваивается значение $y_i=D$ многозначного алфавита. Если же в исправной схеме $y_i=0$, а в неисправной $-y_i=1$, то присваивается $y_i=D'$. Остальные переменные состояния y_j , которые имеют одинаковые значения в исправной и неисправной схемах получают значения $y_j=0$ или $y_j=1$ соответственно.

Кроме этого, необходимо внести влияние неисправности в каждый комбинационный эквивалент итеративной комбинационной схемы. Для одиночной константной неисправности $x_i=0$ в алфавите B_{16} этой переменной присваивается значение $x_i=F0$. Соответственно для одиночной константной неисправности $x_i=1$ переменной присваивается значение $x_i=F1$. Далее для каждого потенциального решения \tilde{T} – искомой входной последовательности необходимо выполнить логическое моделирование в универсальном многозначном алфавите B_{16} . Если в результате моделирования значения D или D' достигнут хотя бы одного внешнего выхода итеративной комбинационной схемы, то последовательность \tilde{T} различает данную пару состояний. Для организации перебора потенциальных решений – различающих входных последовательностей в процессе поиска мы используем один из самых современных и эффективных методов - генетический

алгоритм [1].

Можно показать, что приведенный выше алгоритм позволяет построить тестовую последовательность $T=0, 0, 0, 0$, которая проверяет неисправность $f_1=0$ схемы рис.1 (относительно только кратной стратегии наблюдения выходных сигналов).

Выше представлен только укрупненный алгоритм. При его реализации некоторые шаги могут быть выполнены по-разному, и здесь есть много нюансов. Выбор следующей пары состояний на шаге 2 алгоритма можно производить различным способом. При этом переменные состояния могут принимать либо полностью, либо частично определенные значения. В целом процесс построения тестовой последовательности дает значительно лучшие результаты (по получаемой полноте теста) по сравнению с классическими алгоритмами, которые используют одиночную стратегию наблюдения сигналов (SOT) и полностью неопределенные начальные состояния как исправной, так и неисправной схем. Здесь можно использовать различные эвристики.

Во-первых, позволять переменным состояниям как исправной, так и неисправной схем принимать определенные значения. Это оправдано в том случае, когда исправную и неисправную схемы трудно инициализировать – установить элементы памяти в определенное состояние. Очевидно, использование полностью определенных состояний позволяет отказаться от требования инициализации схемы.

Во-вторых, позволять принимать определенные значения только переменным состояниям неисправной схемы, оставляя неопределенными переменные состояния исправной схемы. Это целесообразно в том случае, когда исправная схема легко инициализируема, а неисправная – не инициализируема. В этом случае несложно построить инициализирующую последовательность для исправной схемы и использовать ее в качестве начального фрагмента тестовой последовательности, которая переведет ее в определенное состояние. Это позволяет резко сократить число распознаваемых пар состояний – необходимо отличить одно состояние исправной схемы от всех состояний неисправной.

В третьих, позволять принимать определенные значения только переменным состояниям исправной схемы, оставляя неопределенными

значения переменных состояния неисправной схемы. Эта эвристика полезна в том случае, когда можно инициализировать неисправную схему, а исправную нельзя.

Заключение

Следует отметить, что для схем с памятью количество неисправностей, непроверяемых относительно одиночной (обычной) стратегии наблюдения выходных сигналов, может быть достаточно большим. Так, например, для схем каталога ISCAS89 [7] даже для одиночных константных неисправностей число таких неисправностей по некоторым данным [3] в среднем достигает 38%. Поэтому для достижения высокой полноты проверяющих тестов последовательностных схем нельзя ограничиваться только одиночной (обычной) стратегией наблюдения выходных сигналов с использованием трючного моделирования. Эту стратегию можно применять на первой стадии моделирования или генерации тестов. При этом имеет смысл из множества всех неисправностей выделить неисправности, которые в принципе не могут быть проверены с использованием одиночной стратегии наблюдения выходных сигналов. Далее к таким неисправностям следует применять разработанные методы генерации тестов с использованием более точной кратной стратегии наблюдения выходных сигналов, что позволяет существенно повысить полноту обнаружения неисправностей в схемах с памятью.

СПИСОК ЛИТЕРАТУРЫ

1. Ю.А.Скобцов, В.Ю.Скобцов. Логическое моделирование и тестирование цифровых устройств. – Донецк: ИПММ НАНУ, ДонНТУ, 2005. – 436с.
2. M.L.Bushnell, V.D.Agrawal. Essentials of electronic testing for digital, memory and mixed-signal VLSI

circuits. – Kluwer academic publishers, 2001. – 690p.

3. I. Pomeranz and S.M. Reddy, "The multiple observation time strategy". // *IEEE Transactions on Computers*. – 1992. – Vol. 41, No. 5. – P.627-637.
4. Ю.А.Скобцов, Д.В.Сперанский Аналитический метод построения различающих последовательностей для дискретных устройств // *Автоматика и телемеханика*. – 1980. – №1. – С.122-130.
5. V.Becker, M.Keim. Hybrid fault simulation for synchronous sequential circuits // *Journal of electronics: Theory and applications*. – 1999. – №15. – P.219-238.
6. Ю.А.Скобцов, В.Ю.Скобцов, Ш.Н.Хинди. Символьное моделирование неисправных последовательностных схем на основе кратной стратегии наблюдения выходных сигналов // *Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація*. – Вип.15(139). – Донецьк, 2008. – С.122-127.
7. Brgles F., Bryan D., Kozminski K. Combinational profiles of sequential benchmark circuits // *International symposium of circuits and systems, ISCAS-89*. – 1989. – P.1929-1934

U.Skobtsov, V.Skobtsov, Sh.Hindy **Test generation for sequential machines by using multiple strategy of output signal observation.**

Ю.А.Скобцов, В.Ю.Скобцов, Ш.Н.Хинди **Генерація тестів для послідовних схем з використанням кратної стратегії спостереження вихідних сигналів.**

Розроблено метод побудови тестових послідовностей для цифрових схем з пам'яттю на основі розрізнення пар станів справної на несправній схемі. Використовується кратна стратегія спостереження вихідних сигналів, 16-значній алфавіт та генетичні алгоритми. Запропонований метод дозволяє виявити несправності, що не перевіряються традиційними методами і, тим самим, підвищити повноту контролюючих тестів.