УДК 681.32

Ю.А. СКОБЦОВ[1], В.Ю. СКОБЦОВ[2], Ш.Н. ХИНДИ[1]

[1]Донецкий национальный технический университет, Украина
[2]Институт прикладной математики и механики НАН Украины, Украина

ДВУХУРОВНЕВЫЙ АЛГОРИТМ ГЕНЕРАЦИИ ПРОВЕРЯЮЩИХ ТЕСТОВ ДЛЯ СХЕМ С ПАМЯТЬЮ

Представлен двухуровневый алгоритм генерации тестов, где на нижнем уровне эволюционными методами генерируются входные характеристические последовательности, которые позволяют установить некоторые элементы памяти в определенные состояния и тем самым упростить генерацию тестов. Алгоритм второго верхнего уровня при генерации тестов использует произвольные входные последовательностями и характеристические последовательности, построенные на нижнем уровне ГА, что делает эволюционный поиск более направленным и повышает его эффективность. При оценке полноты тестов используется кратная стратегия наблюдения сигналов.

Ключевые слова: *последовательностные схемы, генерация тестов, генетические алгоритмы, кратная стратегия наблюдения.*

Введение

В диагностике цифровых схем с памятью в последнее время достигнуты значительные успехи, которые, прежде всего, основаны на использовании методов искусственного интеллекта. Сама постановка задачи построения проверяющих тестов для схем с памятью существенно зависит от применяемой стратегии наблюдения выходных сигналов. Известно, что на практике, в основном, используются методы генерации тестов, основанные на логическом моделировании в троичном алфавите с применением одиночной (обычной) стратегии наблюдения выходных сигналов, которые не позволяют точно оценить полноту проверяющих тестов [1,2]. В настоящее время при генерации тестов широко применяются последовательные и параллельные генетические алгоритмы (ГА), которые позволили существенно повысить их качество и быстродействие [1,2].

Целью данной работы является дальнейшее развитие эволюционных алгоритмов построения проверяющих тестов для схем с памятью за счет использования двухуровневых генетических алгоритмов и кратной стратегии наблюдения выходных сигналов.

Основная часть

Постановка задачи построения проверяющего теста для заданной неисправности в последовательностной схеме существенно зависит от используемой стратегии наблюдения неисправностей [1]. 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; $\delta: Y\times X \to Y$ – функция переходов, определяющая следующее состояние автомата; $\lambda: Y\times X \to Z$ – функция выхода, определяющая выходной сигнал. Функции δ и λ реализуются комбинационными схемами, где :

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

$$X=(x_1,...,x_n), \text{ где } x_j=(0,1) \text{ для } j=\overline{1,n}\ ; \quad (2)$$

$$Z=(z_1,...,z_m), \text{ ггд } z_j=(0,1) \text{ для } j=\overline{1,m}. \ .... \quad (3)$$

Обозначим $X(1)$, $X(2)$,…, $X(p)$ – входную последовательность длины *p*. Тогда $Y(Y_0,0)$, $Y(Y_0,1)$,…, $Y(Y_0,p)$ – последовательность состояний автомата, которую он проходит из начального состояния $y_0 \in Y$ под воздействием входной последовательности $X(1)$, $X(2)$,…, $X(p)$. Пусть $Z(Y_0,0)$, $Z(Y_0,1)$,…, $Z(Y_0,p)$ – обозначает выходную последовательность, производимую автоматом из начального состояния Y_0 при подаче входной последовательности $X(1)$, $X(2)$,…, $X(p)$. Обозначим через $z_j(y_0,k)$ для $j=\overline{1,m}$ значение *j*-го выхода на *k*-м шаге моделирования. Используя эти обозначения, следующее состояние определяется следующим образом:

$$Y(y_0,t)=\begin{cases} Y_0 \text{ для } t=0 \\ \delta(X(t),Y(Y_0,t-1)) \text{ для } t\neq 0 \end{cases}. \quad (4)$$

Аналогично выход $Z(y_0,k)$ определяется функцией λ.

Неисправность *f* преобразует автомат *A* в $A=(Y,X,Z,\delta^f,\lambda^f)$, где функции δ^f,λ^f состояния

Y^f и Z^f определяются таким же образом.

Определение. Неисправность f называется обнаружимой в последовательностной схеме входной последовательностью $X(1)$, $X(2)$,…, $X(p)$ относительно стратегии кратного наблюдения выходов, если:

$$\forall(r,q)\exists t \le p, \exists i \le k, \exists b \in \{0,1\}:$$

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

Отметим, что принципиальное отличие между этой стратегией и обычной (одиночного наблюдения выходных сигналов) состоит в следующем. Согласно обычной 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 кратной стратегии для каждой пары состояний исправной и неисправной схемы существует свой момент времени, в котором они дают различные выходные сигналы.

В качестве примера, рассмотрим схему, представленную на рис 1 с одиночной константной неисправностью $f_2 \equiv 0$.

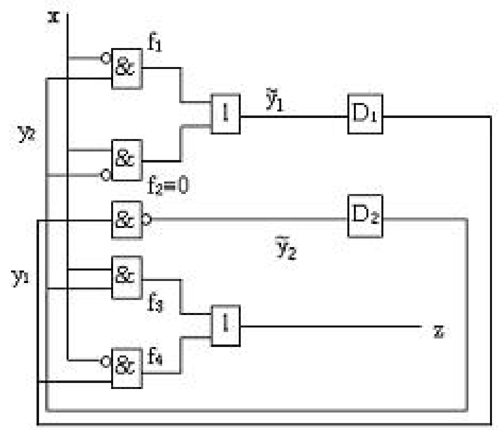

Рис. 1. Пример схемы.

Например, в табл.1 и табл.2 приведены автоматы, реализуемые схемой рис.1, исправной и содержащей неисправность $f_2 \equiv 0$ соответственно.

Таблица 1

Исправный автомат

S	$S_{сл}$, z x=0	$S_{сл}$, z 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

$S(y_1y_2)$	$S_{сл}$, z x=0	$S_{сл}$, z x=1
a	b,0	c,0
b	b,0	b,0
c	a,1	a,1
d	a,1	d,0

Соответственно таблицы 3 и 4 показывают, что входная последовательность X=1, X=1, X=1, X=1 проверяет неисправность $f_2 \equiv 0$ относительно кратной стратегии, поскольку для каждой пары состояний исправной и неисправной схемы существует момент времени, для которого выходные реакции различны. Отметим, что данная неисправность не проверяется этой входной последовательностью относительно одиночной стратегии наблюдения выходных сигналов.

Таблица 3

Реакции исправной схемы

$S(y_1y_2)$	$x_1=1$	$x_2=1$	$x_3=1$	$x_4=1$
A(00)	0	0	1	1
B(01)	0	1	1	0
C(10)	1	0	0	1
D(11)	1	1	0	0

Таблица 4

Реакции неисправной схемы

$S(y_1y_2)$	$X_1=1$	$x_2=1$	$x_3=1$	$x_4=1$
a(00)	0	0	0	0
b(01)	0	0	0	0
c(10)	1	0	0	0
d(11)	1	0	0	0

Естественно кратная стратегия требует больших вычислительных ресурсов, но позволяет повысить полноту тестов.

Для достижения высокой полноты проверяющих тестов последовательностных схем нельзя ограничиваться только одиночной (обычной) стратегией наблюдения выходных сигналов с использованием троичного 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 методы моделирования (или генерации тестов) с использованием более точных стратегий наблюдения выходных сигналов.

В настоящее время существуют два основных подхода к решению этой проблемы. При первом подходе [2] каждая пара состояний исправной и неисправной схемы (что необходимо при кратной стратегии наблюдения выходных сигалов) обрабатывается индивидуально и полученные результаты «склеиваются». Второй подход основан на использовании символьного моделирования исправных и неисправных последовательностных схем [4, 5].

Построение теста на функциональном уровне. Следуя [2] на функциональном уровне (с использованием таблиц переходов автоматов A и A_f, реализуемых исправной и неисправной схемами) общий алгоритм построения проверяющей последовательности X можно представить следующим образом.

1. Установить $X = \varnothing$.

2. Выбрать пару состояний R/F автоматов A и A_f, которая еще не проверена. Если такой пары состояний нет, то X является тестовой последовательностью.

3. С помощью моделирования получить выходные реакции Y, Y_f и следующие состояния S, S_f исправной и неисправной схем для пары состояний R/Q при подаче последовательности X.

4. Если выходные реакции различны, то переход на п.2 (T проверяет данную пару состояний).

5. Найти тестовую последовательность X_n, для начальных состояний S, S_f.

6. Выполнить конкатенацию $X = X \bigcup X_f$ и перейти на п.2.

Отметим, что здесь задана лишь общая схема и на шаге 5 можно использовать любой известный алгоритм построения проверяющей последовательности для заданной пары состояний. Данный ункциональный подход с использованием таблиц переходов автоматов применим только для небольших последовательностных схем, но его можно распространить на структурный уровень, где используется непосредственно логическая схема, а не таблица переходов автомата. Далее для построения проверяющих тестов мы будем использовать структурную модель в виде итеративной комбинационной схемы, кратную стратегию МОТ и многозначный алфавит B_{16}={∅, 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 [1]. При этом анализ двух схем (иправной и неисправной) в двоичном (или троичном) алфавите заменяется анализом одной схемы в многозначном алфавите.

Характеристические последовательности. На нижнем уровне с помощью ГА генерируются характеристические подпоследоватедьности, которые устанавливают значения сигналов для некоторых элементов памяти в определенные значения. Для построения тестов (на верхнем уровне) полезными являются следующие входные характеристические последовательности.

1)*S-последовательность*, которая устанавливает *i*-й триггер в единичное состояние.

2) *R-последовательности,* которая сбрасывает *i*-й триггер в нулевое состояние.

3) Различающая последовательность для *i*-го триггера определяется как входная последовательность, которая производит различные выходные реакции исправной схемы для двух различных состояний, отличающихся значением сигнала на i-ом триггере при неопределенных значениях остальных триггеров.

Представленные характеристические последовательности генерируются с использованием генетического алгоритма нижнего уровня, где в качестве особи используются двоичные последовательности (таблицы) и проблемно ориентированные генетические операторы, описанные в [1,3].

Двухуровневый алгоритм генерации тестов

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

Генерация тестовой последовательности(схема,неисправность)

{

X = ∅, SI = ∅;

While(есть неразличимые пары состояний)

{

Выбор пары (возможно не полностью определенных)состояний(S,Q)

($S = (\alpha_1, \alpha_2, ..., \alpha_k), Q = (\beta_1, \beta_2, ..., \beta_k)$ *),*

не покрываемых на текучий монент SI;

If(неразличимой пары не существует)

Then тест X построен: return;

Формирование значений переменных состояний $(\delta_1, \delta_2, ..., \delta_k)$ *в алфавите* B_{16}*;*

логическое моделирование в алфавите B_{16} *на входной последовательности X с начальными состояниями* $(\delta_1, \delta_2, ..., \delta_k)$ *;*

if (значение D или D' достигает внешнего выхода схемы)

then

пара состояний (S,Q) различается текущей последовательностью X;

Else

{
построение различающей последовательности T (S,Q);
// испольязуются характеристические последовательности нижнего уровня
if(различающая последовательность для (S,Q) не построена)
then
текущая неисправность непроверяемая и удаляется из дальнейшего рассмотрения;
переход на конец алгоритма;
}
Внесение и обработка неопределенностей в переменных состояния();
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 неизбыточной 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 псевдовходам первого комбинационного эквивалента, соответствующим переменным состояния, присваиваются начальные значения, которые определяются парой состояния (S/Q). Если i-я переменная в исправной схеме $y_i = 1$, а неисправной – $y_i^{н} = 0$, то этой переменной присваивается значение y_i=D многозначного алфавита с. Если же в исправной схеме $y_i = 0$, а неисправной – $y_i^{н} = 1$, 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 $x_i \equiv 0$ в алфавите B_{16} этой переменной присваивается значение $x_i \equiv F0$. Соответственно для одиночной константной неисправности $x_i \equiv 1$ переменной присваивается значение $x_i \equiv F1$. 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].

Функция «*Внесение и обработка неопределенностей в переменных состояния()*» позволяет ускорить анализ пар состояний путем внесения неопределенности. При этом аналіз пар состояний исправной и неисправной схем в двоичном алфавите заменяется анализом двух схем в многозначном алфавите. Следует отметить, что использование универсального 16-значного алфавита и одной схемы по сравнению с [2] анализом по отдельности двух (исправной и неисправной) схем в троичном алфавите позволяет существенно уменьшить сложность проверки (скорее всего, свести к линейной!!!).

Внесение и обработка неопределенностей в переменных состояния();

{
Для пары состояний
$S = (\alpha_1, \alpha_2, ..., \alpha_k), Q = (\beta_1, \beta_2, ..., \beta_k)$
For j=1 to k do $\delta_j = F\beta_j$;
логическое моделирование в B_{16};
if(T не различает (S,Q))
then восстановление прежнего значения δ_j;
$\delta_j = G\alpha_j$;
логическое моделирование в B_{16};
if(T не различает (S,Q))
then восстановление прежнего значения δ_j;
end
}

Можно показать, что приведенный выше алгоритм позволяет построить тестовую последовательность T=1, 1, 1, 1, которая проверяет неисправность $f_2 \equiv 0$ схемы рис.1(относительно только кратной стратегии наблюдения выходных сигналов). Здесь используется генетический алгоритм, представленный в [1] с некоторыми модификациями. Во-первых, в начальную популяцию (и в процессе генерации) включаются не только случайные входные последовательности, но и характеристические последовательности, построенные на нижнем уровне. Во-вторых, расширен набор генетических операторов, которые используются при генерации новых особей. В него включены следующие генетические операторы.

Приведенный алгоритм может быть существенно упрощен в том случае, когда (хотя бы исправный) автомат имеет синхронизирующую последовательность (СП). В этом случае СП устанавливает автомат из произвольного состояния в некоторое определенное, которое нужно отличить от состояний неисправного автомата. Более того, как правило в этом случае, можно ограничиться стратегией одиночного наблюдения выходных сигналов

Выводы

Следует отметить, что для последовательностных схем количество неисправностей, непроверяемых относительно одиночной стратегии наблюдения выходных сигналов, может быть достаточно большим. Так например, для схем каталога ISCAS89 даже для одиночных константных неисправностей число таких неисправностей по некоторым данным [2] в среднем достигает 38%. Предложенный двухуровневый алгоритм построения проверяющих тестов с использованием характеристических последовательностей и стратегии кратного наблюдения выходных сигналов позволяет существенно повысить полноту проверяющих тестов для схем с памятью, но требует значительных вычислительных ресурсов.

Литература

1. Скобцов Ю.А. Логическое моделирование и тестирование цифровых устройств / Ю.А. Скобцов, В.Ю. Скобцов. – Донецк: ИПММ НАНУ, ДонНТУ, 2005. – 436 с.

2. Pomeranz I. The multiple observation time strategy / I.Pomeranz, S.M. Reddy // IEEE Transactions on Computers. – 1992. – Vol. 41. – N 5. – P. 627-637.

3. Скобцов Ю.А. Иерархические эволюционные алгоритмы построения проверяющих тестов цифровых последовательностных схем / Ю.А. Скобцов, В.Ю. Скобцов, Ш.Н. Хинди // Радіоелектронні і комп'ютерні системи. – ХАІ. – 2008. – №6. – С. 159-163.

4. Becker B. Hybrid fault simulation for synchronous sequential circuits / B.Becker,M.Keim // Journal of electronics: Theory and applications. –1999. – № 15. – P. 219-238.

5. Скобцов Ю.А. Символьное моделирование неисправных последовательностных схем на основе кратной стратегии наблюдения выходных сигналов / Ю.А. Скобцов, В.Ю. Скобцов, Ш.Н. Хинди // Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація, 2008 – Вип.15(139). – С. 122-127.

Поступила в редакцию 28.01.2009

Рецензент: д-р техн. наук, проф. А.В. Дрозд, Одесский национальный политехнический университет, Одесса, Украина.

ДВОРІВНЕВИЙ АЛГОРИТМ ГЕНЕРАЦІЇ ПЕРЕВІРЯЮЧИХ ТЕСТІВ ДЛЯ СХЕМ З ПАМ'ЯТЬЮ

Ю.О. Скобцов, В.Ю. Скобцов, Ш.Н. Хінді

Представлено дворівневий алгоритм генерації тестів, де на нижньому рівні еволюційними методами спочатку генеруються вхідні характеристичні послідовності, які дозволяють встановити деякі елементи пам'яті в певні стани й тим самим спростити генерацію тестів. Алгоритм другого верхнього рівня при генерації тестів використовує довільні вхідні послідовності та характеристичні послідовності, які побудовані на нижньому рівні ГА, що робить еволюційний пошук більш спрямованим та підвищує його ефективність. Під час оцінки повноти тестів використовується кратна стратегія спостереження сигналів.

Ключові слова: послідовні схеми, генерація тестів, генетичні алгоритми, кратна стратегія спостереження.

TWO-LEVEL ALGORITHM OF TEST PATTERN GENERATION FOR CIRCUITS WITH MEMORY

Yu.A. Skobtsov, V.Yu. Skobtsov, Sh.N. Hindi

There was represented two-level algorithm of test pattern generation where at the low level firstly some input characteristic sequences are generated with the evolutionary methods, which allow to set some elements of l memory to certain states and to reduce test generation time. The algorithm of second high level uses for test generation arbitrary input sequences and characteristic sequences generated at low level. It makes evolutionary search more directed and increases his effectiveness. During test coverage evaluation the multiple observation time strategy is used.

Key words: sequential schemes, test pattern generation, genetic algorithms, multiple observation time strategy.

Скобцов Вадим Юрьевич – канд. техн. наук, доцент, Институт прикладной математики и механики НАН Украины, Донецк, Украина. skobtsov@iamm.ac.donetsk.ua

Скобцов Юрий Александрович – д-р техн. наук, проф., зав. кафедрой, Донецкий национальный технический университет, Донецк, Украина. skobtsov@kita.dgtu.donetsk.ua

Хинди Шукри Насри – аспирант, Донецкий национальный технический университет, Донецк, Украина. Shukri_ua@hotmail.com.