

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ОПТИМИЗАЦИИ РАССЕИВАНИЯ ТЕПЛОВОЙ ЭНЕРГИИ ВХОДНЫХ ТЕСТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Иванов Д.Е., к.т.н., с.н.с., доцент, Институт Прикладной Математики и
Механики НАН Украины,
E-mail: ivanov@iamm.ac.donetsk.ua

Abstract

In this paper a new approach for solving the problem of the optimization of the power dissipation under test sequence application is proposed. This approach is based on the redundancy of the test sequences and consists of the steps: redundant test generation, evaluating power dissipation for generated test sequences and construction of the subset of sequences with optimal parameters. The last stage is based on the genetic algorithm. Also we give the results of the computer experiments on the ISCAS-89 benchmark circuits that show the effectiveness of the proposed approach.

1. Введение

Широкое распространение для решения различных задач технической диагностики получили генетические алгоритмы (ГА) [1-2]. Это связано с тем фактом, что классические детерминированные методы решения оказываются неэффективными в условиях продолжающегося роста размерности дизайна цифровых схем. В частности, ГА эффективно применяются при решении задачи построения входных идентифицирующих последовательностей цифровых устройств (ЦУ): тестовых [3-4], инициализирующих [5] и последовательностей, проверяющих эквивалентность [6].

Основным недостатком классических методов, которые основаны на преобразовании булевого представления схемы [7] или построения дерева решения [8], является переполнение памяти, что делает их практически неприменимыми к реальному размеру современных цифровых устройств. Это связано с тем, что полное представление решения в памяти инструментальной ЭВМ невозможно из-за размера дерева решений, которое в свою очередь определяется числом элементов состояний в ЦУ. ГА сумели обойти данное ограничение благодаря тому, что они не строят решение в явном виде. Эта задача, фактически, заменена на две другие: моделирование и основанная на нём оценка качества потенциального решения. При этом для задач построения входных последовательностей ГА представляются как направленные стохастические методы поиска решений, основанные на моделировании. Для оценки потенциальных решений используются как алгоритмы исправного моделирования, так и алгоритмы моделирования с неисправностями. Задача моделирования считается решённой для последовательностных схем. Классическим инструментом, применяемым для этих целей, является алгоритм моделирования, предложенный в [9], а также его различные модификации [10-11].

Недостатком алгоритмов построения входных последовательностей, которые основаны на ГА, является существенно большая длина получаемых решений в сравнении с детерминированными подходами [12].

Применение ГА не ограничивается лишь только построением входных идентифицирующих последовательностей. Большое количество задач технической диагностики могут быть решены с помощью комбинаторного перебора. Размерность таких

задач велика, поэтому эффективно решать их могут генетические алгоритмы, поскольку они обладают наилучшими нелокальными поисковыми свойствами [2].

В последнее время актуальность приобретает цифровая техника, которая обладает минимальным потреблением электрической энергии. Данное требование обусловлено как потребительскими предпочтениями рынка электроники, так и общими усилиями человечества к снижению потребления любого вида энергии. С другой стороны, потребление энергии во время тестирования существенно выше, чем при обычной работе ЦУ [13], поэтому режим тестирования является критическим с этой точки зрения.

Для решения задачи минимизации потребления энергии во время приложения теста предложено несколько подходов. Например, в [14] для этих целей адаптируется известный алгоритм PODEM. При этом присвоение значений на неопределённых линиях происходит таким образом, чтобы минимизировать число событий. В работе [15] предложен метод минимизации рассеивания тепла для заданного теста, что достигается упорядочиванием входных наборов.

В данной статье предлагается иной подход к решению задачи. Он состоит из 3 этапов и основан на генерации множества избыточных тестовых последовательностей. После генерации происходит оценка параметра рассеивания тепла полученных последовательностей, а затем выбор оптимального подмножества. Данный метод, в отличие от ранее предложенных, также применим к нециализируемым схемам, в которых отсутствуют цепи сброса элементов состояний. При этом никакие изменения в саму схему не вносятся.

Данная статья имеет следующую структуру. Во введении обоснована актуальность исследования и связь с другими работами в этом направлении. В первом разделе описывается формальная постановка задачи и этапы общего подхода её решения. Следующий раздел описывает генетический алгоритм выбора оптимального подмножества входных последовательностей. В третьем разделе описана программная реализация алгоритма и приведены результаты численных экспериментов на контрольных схемах. В заключении делаются выводы и указываются направления возможных дальнейших исследований.

2. Общий подход решения задачи

В качестве модели в данной работе используются синхронные последовательностные схемы с нулевой задержкой распространения сигнала и физической реализацией по КМОП технологии.

Целью работы является разработка алгоритма, который строит входные тестовые последовательности с минимальным рассеиванием тепла.

Решение указанной задачи осуществляется в три этапа:

- генерация избыточных по покрытию неисправностей входных последовательностей;
- оценка качества решения задачи для каждой из таких последовательностей;
- поиск оптимального в смысле целей задачи подмножества входных последовательностей.

Первые два этапа являются подготовительными, а третий является задачей комбинаторной оптимизации и решается в нашем подходе с помощью ГА.

Для формальной постановки задачи дадим определение избыточности входной тестовой последовательности.

Определение 1. Неисправность $f \in \tilde{F}_{const}$ обнаруживается (проверяется) заданной последовательностью $S = \{s_1, s_2, \dots, s_l\}$ с избыточностью r , если существует не менее чем r подпоследовательностей $s_{i_1}, s_{i_2}, \dots, s_{i_r} \in S$, где $l \geq r$, таких, что каждая из них проверяет неисправность f . При этом r называется фактором избыточности. Здесь \tilde{F}_{const} - сжатое по эквивалентности множество одиночных константных неисправностей.

Определение 2. Входная тестовая последовательность s_{ex} обладает полной избыточностью r , если каждая неисправность класса \tilde{F}_{const} проверяется данной последовательностью не менее r раз.

Определение 3. Избыточная полнота последовательности s_{ex} при избыточности r равна

$$P_r(s_{ex}) = \frac{m_r}{|\tilde{F}_{const}|}, \text{ где } m_r = |\tilde{F}_{const}^r(s_{ex})| - \text{ число неисправностей из множества } \tilde{F}_{const},$$

которые проверяются не менее r раз заданной входной последовательностью.

Этап 1 решения задачи заключается в построении входной последовательности (набора подпоследовательностей) $S = \{s_1, s_2, \dots, s_l\}$, которая обладает требуемой избыточной полнотой при фиксированной избыточности r . При этом для каждой фиксированной неисправности из $\tilde{F}_{const}^r(s_{ex})$ гарантируется существование не менее r входных подпоследовательностей, принадлежащих S таких, что каждая из них обнаруживает данную неисправность. Гарантирование избыточности для некоторой неисправности позволяет в дальнейшем в зависимости от целей задачи выбирать в качестве окончательного решения ту или иную из последовательностей.

Решение задачи построения избыточной тестовой последовательности возможно путём адаптации практически любого генератора тестов для последовательностных схем. Мы адаптировали для этих целей генетический алгоритм построения тестов, разработанный ранее авторами [4]. В начальном варианте алгоритма неисправность удаляется из общего списка непроверенных неисправностей сразу, когда будет обнаружена очередной строящейся последовательностью. В модифицированном варианте алгоритма неисправность будет вычеркнута из списка целевых неисправностей в том случае, если уже обнаружена r раз, где r - фактор избыточности.

Этап 2 заключается в предварительной оценке подпоследовательностей, которые входят в S : для каждой подпоследовательности $s_i \in S$ вычисляется числовая оценка $E(s_i)$, которая показывает рассеивание тепловой энергии при моделировании работы схемы на заданной входной последовательности. Задача фактически заключается в моделировании работы схемы на заданной входной последовательности и получении на его основании необходимых данных.

Показатель рассеиваемой тепловой энергии $E(s_i)$ при моделировании заданной входной последовательности s_{ex} существенно зависит от технологии физической реализации микросхем. Наиболее распространённой в настоящее время технологией производства является КМОП. Активность моделируемой схемы, вызванная приложением заданной входной последовательности, влияет только на динамическую составляющую рассеиваемой энергии, тогда как статическая часть никак от неё не зависит. Также известно, что доминирующей является динамическая составляющая рассеивания тепла.

Известно [16], что для КМОП технологии рассеиваемая мощность для заданной входной последовательности s_{ex} определяется выражением

$$E(s_{ex}) = 0,5 \cdot V^2 \cdot C \cdot f \cdot A(s_{ex}) \quad (1)$$

где: V - напряжение работы схемы, C - физическая ёмкость на выходе вентиля, f - частота работы схемы, $A(s_{ex})$ - число событий при моделировании на заданной входной последовательности.

Таким образом, для оценки рассеиваемой тепловой мощности с точностью до постоянного коэффициента нам необходимо вычислить параметр $E(s_{ex})$, который в свою очередь имеет следующее представление:

$$A(s_{ex}) = \sum_{i=1}^L \sum_g A_g^i, \quad (2)$$

где:

L - длина входной последовательности (число входных наборов), а активность вентилей определяется выражением

$$A_g^i = \begin{cases} 1, & \text{если выход вентилей } g \text{ изменился при моделировании на наборе с номером } i; \\ 0, & \text{в противном случае.} \end{cases} \quad (3)$$

Видно, что рассеиваемая мощность прямо пропорциональна числу событий, причём все остальные переменные являются фиксированными для заданного режима работы схемы. Произвести вычисление параметра $A(s_{ex})$ позволяет сделать любой алгоритм событийного моделирования работы неисправных схем. Авторами для данной задачи описанный ранее алгоритм [11] с нулевыми задержками распространения сигнала.

3. Генетический алгоритм выбора множества подпоследовательностей.

В данном разделе мы подробно опишем алгоритм решения задачи третьего этапа. Он основан на генетическом алгоритме как средстве комбинаторной оптимизации.

Этап 3 заключается в выборе подмножества последовательностей $S' \subset S$ такого, что выполняются два следующих условия:

$$P(S') = P(S), \quad (4)$$

и

$$E(S') \rightarrow \min. \quad (5)$$

Рассмотрим пример. Пусть для некоторой схемы множество проверяемых неисправностей $\tilde{F}_{const} = \{f_1, f_2, \dots, f_{10}\}$, тестовая последовательность $S = \{s_1, s_2, \dots, s_6\}$. Подпоследовательности $s_i, i = \overline{1,6}$ имеют следующие характеристики:

s_1 : рассеиваемая тепловая мощность 45, проверяемые неисправности: 1, 3, 5, 7, 9, 10;

s_2 : рассеиваемая тепловая мощность 15, проверяемые неисправности: 1, 3, 5, 10;

s_3 : рассеиваемая тепловая мощность 13, проверяемые неисправности: 2, 4, 6;

s_4 : рассеиваемая тепловая мощность 30, проверяемые неисправности: 1, 2, 5, 7, 8, 10;

s_5 : рассеиваемая тепловая мощность 10, проверяемые неисправности: 6, 7, 8, 10;

s_6 : рассеиваемая тепловая мощность 15, проверяемые неисправности: 1, 2, 8, 9, 10;

Пусть в качестве возможных решений выступают следующие подмножества S' :

$S_1 = \{s_1, s_3, s_5\} \subset S$, $S_2 = \{s_1, s_3, s_6\} \subset S$, $S_3 = \{s_2, s_3, s_5\} \subset S$ и $S_4 = \{s_1, s_3, s_4, s_6\} \subset S$. Рассчитаем характеристики данных последовательностей:

S_1 : рассеиваемая тепловая мощность 68, полнота теста $P(S_1) = 100\%$;

S_2 : рассеиваемая тепловая мощность 73, полнота теста $P(S_2) = 100\%$;

S_3 : рассеиваемая тепловая мощность 38, полнота теста $P(S_3) = 90\%$;

S_4 : рассеиваемая тепловая мощность 103, полнота теста $P(S_4) = 100\%$;

Из расчётов видно, что из множества потенциальных решений сразу необходимо исключить набор S_3 , поскольку он обладает худшими тестовыми свойствами относительно других наборов и не обеспечивает выполнения первого из условий. Из наборов с наилучшими тестовыми свойствами необходимо выбрать тот, который обладает наименьшим параметром рассеивания, т.е. S_1 . Также из примера видно, что набор S_4 является избыточным, и из него без потери полноты можно исключить либо последовательность s_4 (рассеивание тепла = 73), либо последовательность s_6 (рассеивание тепла = 88). Хотя в этих случаях полнота теста останется равной 100%, параметр

рассеивания мощности окажется больше, чем у других наборов, и данное множество также не будет выбрано в качестве оптимального.

Оценим практическую сложность задачи. Если входная последовательность S состоит из n подпоследовательностей, то число возможных подмножеств 2^n . Например для контрольной схемы средней размерности s1488 из каталога ISCAS-89 на этапе 1 сгенерирована последовательность S , которая состоит из 356 подпоследовательностей, т.е. $n = 356$. В этом случае число возможных множеств подпоследовательностей будет $2^n = 2^{356} \approx 1,47e107$. Перебор такого числа вариантов является нереальным для современных рабочих станций. Поэтому для решения данной задачи предлагается использовать генетический алгоритм.

Известно [2], что для построения генетического алгоритма необходимо задать его компоненты: вид особей и популяций, оценочную функцию, генетические операции. Авторы неоднократно описывали сам генетический алгоритм и его применение к решению задач диагностики [4-6], поэтому здесь для краткости изложения его описание опущено. Ограничимся только описанием компонент, которые необходимы для понимания сути описываемого здесь алгоритма.

В качестве особи в предлагаемом алгоритме выступает множество номеров подпоследовательностей из S . Номера в разных особях изменяются в ходе эволюции, также как и их число в различных особях. Мы не учитываем порядок номеров в особи.

Операция скрещивания выполняется над двумя родителями и порождает одного потомка. При этом число номеров подпоследовательностей в нём будет равно $n/2$, где n - длина первого родителя. В список потока включаются следующие номера подпоследовательностей: первый $n/2$ случайно копируются из первого родителя, оставшиеся – из второго.

Используются три «классические» операции мутации для множеств: обмен элементами, удаление элемента и добавление случайного элемента. При этом выбор между типами операций мутации происходит также случайно.

Вид оценочной функции существенно влияет на поисковые свойства всего алгоритма, а также на скорость его работы. Работа ГА по выбору оптимального подмножества последовательностей состоит из двух фаз, каждой из которых соответствует своя оценочная функция.

В первой фазе из множества S выбираются такие подмножества, для каждого из которых выполнено условие $P(S') = P(S)$, т.е. подпоследовательности, входящие в множество S' должны обнаруживать все те неисправности, которые обнаруживают последовательности из S . Таким образом цель первой фазы ГА – увеличение показателя полноты теста для всех особей в популяции до уровня $P(S)$. Для особи с номером i оценочная функция будет иметь вид:

$$F_i = P(S_i) - P(S). \quad (6)$$

После того как будет выполнено условие (4) для всех особей в популяции, ГА переходит в фазу 2, целью которой является уменьшение рассеивания тепла для каждого подмножества, но без потери тестовых свойств. При этом, если в результате генетических операций (скрещивание и мутация) будут построены особи, которые не удовлетворяют условию (4), то они не рассматриваются далее. Таким образом, для фазы 2 ГА в качестве оценочной функции выступает выражение:

$$F_i = E_{\max} - E_i. \quad (7)$$

где E_{\max} - показатель рассеивания тепла для множества S .

При построении новой популяции используется стратегия элитизма.

4. Экспериментальные данные.

Для алгоритмов трёх этапов решения задач авторами была произведена их программная реализация.

Для первого этапа построения избыточных тестовых последовательностей был адаптирован описанный ранее авторами генетический алгоритм построения тестов [4]. Изменения касались того фрагмента кода, в котором определяется: вносить ли влияние неисправности для моделирования на данном входном наборе [11]. В исходной версии некоторая неисправность f_i вносилась в список для моделирования, если:

- она была не обнаружена к текущему моменту времени;
- она являлась активной, т.е. при моделировании для данной неисправности на предыдущем такте модельного времени имелись различия на псевдовыходах исправной и неисправной схем.

В модифицированной версии алгоритма для внесения неисправности на моделирование в первом из указанных условий мы установили дополнительное требование, чтобы неисправность была проверена менее r раз, где r - параметр избыточности.

Также исходный генератор тестов строил одну последовательность, не фиксируя в ней отдельные подпоследовательности. Наш алгоритм использует эту существенную информацию. Поэтому в структуру данных, которая хранит тестовые наборы, было внесено изменение, которое для тестовой последовательности фиксирует такты времени, в которые начинаются подпоследовательности.

Для второго этапа был адаптирован ранее предложенный авторами алгоритм событийного моделирования цифровых схем с неисправностями [11]. Изменения заключались в следующем. Для каждой подпоследовательности моделирование начиналось из неопределённого состояния схемы и фиксировалось число событий при моделировании на заданной входной последовательности. В результате работы строился новый файл, который содержал следующую информацию:

- общее число подпоследовательностей;
- общее число неисправностей в сжатом списке;
- далее для каждой подпоследовательности: длина, число событий моделирования, список, содержащий номера проверяемых неисправностей.

Данный файл являлся исходным для программы, которая реализует третий этап подхода - оптимизацию.

Программная реализация ГА третьего этапа составила примерно 2000 строк кода на языке C++.

Эффективность любой реализации ГА существенно зависит от большого числа эвристик. Покажем порядок исследования эффективности ГА от различных параметров. Для проведения машинных экспериментов была выбрана схема средней размерности s1488. Все эксперименты проводились при факторе избыточности $r = 5$.

Вначале исследовалась схема построения новой популяции из предыдущей и промежуточной. Рассматривались две схемы. Первая – «классическая». Ниже представлен фрагмент псевдокода такой схемы, который отвечает за построение промежуточной популяции.

```
для( i=0 ; i<IndividualsNumber ; i++ )
{
    ОперацияВыбора (Популяция, Родитель1, Родитель2);
    НоваяОсобь=Срещивание (Родитель1, Родитель2);
    НоваяОсобь=Мутация (НоваяОсобь);
    ДобавитьВПромежуточнуюПопуляцию (НоваяОсобь);
}
```

На рис.1 представлен рост эффективности ГА в зависимости от числа поколений (ломаная кривая). Неустойчивость этого роста заставила предложить «альтернативную» схему. Псевдокод построения промежуточной популяции для данной схемы приведён ниже.

```

для( i=0 ; i<IndividualsNumber/2 ; i++ )
{
    ОперацияВыбора(Популяция, Родитель1, Родитель2);
    НоваяОсобь=Срещивание(Родитель1, Родитель2);
    ДобавитьВПромежуточнуюПопуляцию(НоваяОсобь);
    ОперацияВыбора(Популяция, Родитель1);
    НоваяОсобь=Мутация(Родитель1);
    ДобавитьВПромежуточнуюПопуляцию(НоваяОсобь);
}

```

Рост оценочной функции в зависимости от поколения представлен на рис.1 (плавная кривая). Таким образом, для дальнейших исследований была выбрана вторая схема.

При выбранной схеме реализации ГА необходимо определить значения параметров «число особей в популяции» и «максимальное число поколений», тогда как определять численные значения вероятностей мутации и скрещивания нет необходимости.

Выбор числа особей в популяции. Для определения параметра числа особей в популяции был произведён ряд экспериментов, в которых значение параметра изменялось от 10 до 100 с шагом 10. Как результирующий фиксировался параметр уменьшения рассеивания тепловой мощности полученной тестовой последовательности в сравнении с исходной. Результаты усреднялись по 10-и экспериментам.

График полученной зависимости изображён на рис.2. Видно, что монотонный рост происходит в экспериментах с числом особей в популяции до 70-и включительно. Далее происходит обвал графика. Такой скачок объясняется тем, что при увеличении числа особей в популяции алгоритм не всегда переходит к фазе 2 оптимизации. Это связано с условием этого перехода: он происходит в тот момент, когда все особи в

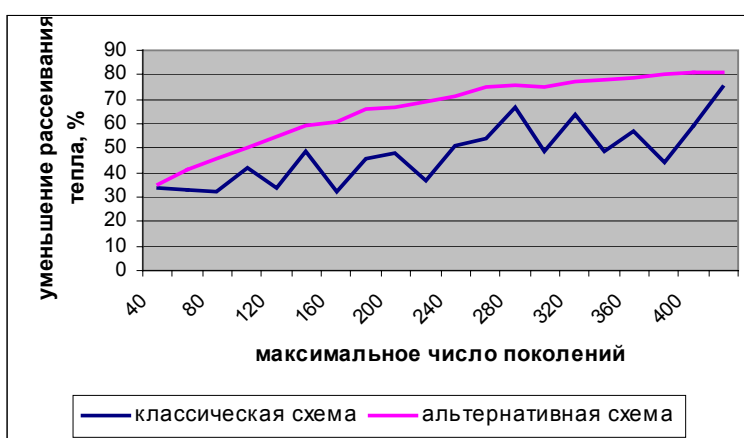


Рис.1 Исследование схемы ГА.

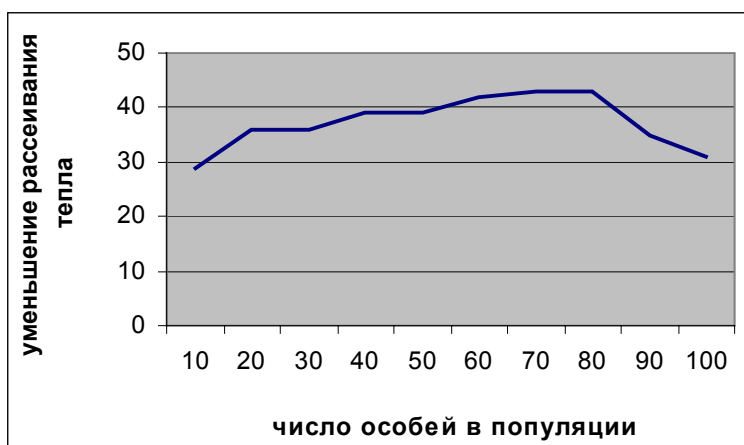


Рис.2. Зависимость эффективности ГА от числа особей в популяции.

популяции обладают полнотой, которая соответствует исходной последовательности: $P(S') = P(S)$. При росте числа особей выполнить это условие становится труднее и при числе особей в популяции более 70 алгоритм при некоторых запусках даже не переходит в фазу 2 уменьшения рассеивания тепла. Причём, чем больше особей в популяции, тем больше вероятность такого негативного события. Таким образом, в качестве число особей в популяции выбрано значение 70.

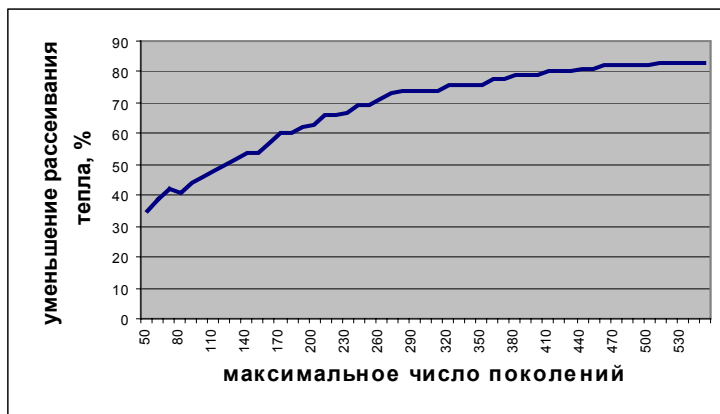


Рис.3. Эффективность ГА в зависимости максимального от числа поколений.

Выбор числа поколений в генетическом алгоритме. Для определения числового значения данного параметра также был проведён ряд экспериментов: значение изменялось от 100 до 495 с шагом 10. Усреднение производилось по 20-и экспериментам. График зависимости снижения рассеивания тепловой мощности от максимального числа поколений в ГА приведён на рис.3. Видно, что рост графика прекращается при числе поколений более 500. Данное значение и выбрано для дальнейших экспериментов.

С определёнными таким образом значениями были проведены машинные эксперименты с другими схемами каталога ISCAS-89 [17]. Их результаты приведены в табл.1.

Из приведённых числовых данных можно сделать следующие выводы:

- наилучшие значения показателя снижения рассеивания тепла достигаются для легкотестируемых схем: s3271, s2081, s1494 и т.д.;
- даже для труднотестируемых схем данный показатель превышает 70%;
- время, которое затрачивается на этап 3 «выбор оптимального подмножества», является существенно меньшим в сравнении с этапом 1 «генерация избыточного теста».

5. Заключение.

В статье предложен подход построения тестовых последовательностей с минимальным рассеиванием тепловой энергии. Подход состоит из трёх последовательных этапов: построение избыточной тестовой последовательности, оценка рассеивания тепла для каждой подпоследовательности, выбор оптимального множества подпоследовательностей. Для решения задачи третьего этапа предложено использовать генетический алгоритм. Эффективность предложенного подхода демонстрируется на контрольных схемах ISCAS-89, при этом среднее уменьшение рассеивания тепла составило 85.6%, достигая 93.25% в лучших случаях.

Поскольку внедрение различных средств оптимизации непосредственно в алгоритмы генерации тестов ведёт к существенному росту временных затрат при решении задачи [12], то данный трёхэтапный подход может быть применён к другим задачам оптимизации входных диагностических последовательностей. В частности, можно отметить возможность применения данного подхода к задаче уменьшения длины тестов.

Литература.

1. Goldberg D.E., “Genetic Algorithm in Search, Optimization, and Machine Learning”, Addison-Wesley.- 1989.- 432p.
2. Ю.А. Скобцов. Основы эволюционных вычислений.– Донецк: ДонНТУ, 2008.- 326с.

Табл.1. Экспериментальные результаты работы программы.

схема	полнота теста, %	уменьшение рассеивания тепла, %	длина последовательности		время работы, час:мин:сек.	
			начальная	достигнутая	общее	ГА
s298	85.71	90.77	1467	166	0:21	0:00
s344	96.19	88.30	3621	550	1:01	0:00
s349	95.42	91.04	1960	197	0:27	0:00
s382	89.47	86.46	8661	1387	3:36	0:00
s386	73.95	89.23	17826	1959	5:34	0:00
s400	88.21	92.51	16617	1546	10:43	0:00
s444	79.32	87.32	3676	524	3:00	0:00
s526	8.65	86.09	70	10	0:16	0:00
s635	0.15	73.51	40	10	53:00	0:00
s641	44.75	88.91	57372	7134	1:28:25	0:00
s713	81.93	89.02	11841	1532	3:35	0:00
s832	48.05	89.21	3135	351	1:58	0:01
s938	4.40	86.20	1952	16	1:45:26	0:00
s967	7.41	89.56	190	20	1:05:13	0:00
s1196	91.22	76.48	16117	3596	1:16	0:24
s1238	77.86	74.92	7025	1902	0:56	0:07
s1269	17.87	79.75	647	134	2:55	0:00
s1423	73.47	80.78	76075	15442	2:01:26	0:10
s1488	92.19	82.44	16968	3342	6:22	0:14
s1494	95.22	88.32	10966	1327	3:45	0:14
s1512	4.86	84.45	130	20	3:13	0:00
s2081	8.29	93.25	480	32	12:03	0:00
s3271	96.57	86.92	11422	1753	2:15	0:54
s3330	66.45	78.84	5345	1357	16:13	0:09

3. P. Prinetto, M. Rebaudengo, M. Sonza Reorda, "An Automatic Test Pattern Generator for Large Sequential Circuits based on Genetic Algorithms" In Proc. Int. Test Conf., 1994, pp. 240-249.
4. Иванов Д.Е., Скобцов Ю.А. Генерация тестов цифровых устройств с использованием генетических алгоритмов // Труды института прикладной математики и механики НАН Украины. – Т.4. – Донецк, ИПММ. – 1999. – С.82-88.
5. Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб Построение инициализирующих последовательностей синхронных цифровых схем с помощью генетических алгоритмов.- Проблемы информационных технологий, 2007.- №1.- с.158-164.
6. Д.Е. Иванов Генетический подход проверки эквивалентности последовательностных схем // «Радиоелектроніка. Інформатика. Управління».- Запоріжжя, ЗНТУ, 2009.- №1(20).- С.118-123.
7. A. Ghosh, S. Devadas and A.R. Newton, Sequential Logic Testing and Verification, Kluwer Academic Publishers, 1992.
8. O.Coudert, J.C.Madre: "A Unified Framework for Formal Verification of sequential Circuits," ICCAD-90: IEEE Intl. Conf. on Computer Aided Design, Nov. 1990, pp. 134-137
9. Niermann T.M., Cheng W.-T., Patel J.H. PROOFS: A Fast, Memory-Efficient Sequential Circuits Fault Simulator // IEEE Trans. CAD. – 1992.– P.198-207.
10. Kung C.P., Lin C.S. HyHope: A Fast Fault Simulator with Efficient Simulation of Hypertrophic Faults // Proc. of International Test Conference. - 1994. - P.714-718.
11. Иванов Д.Е., Скобцов Ю.А. Параллельное моделирование неисправностей для последовательностных схем // Искусственный интеллект. – 1999. - №1. – С.44-50.
12. F. Corno, M. Sonza Reorda, M. Rebaudengo, Experiences in the use of evolutionary techniques for testing digital circuits // Aproc. of Conf. Applications and science of neural networks, fuzzy

systems, and evolutionary computation, San Diego CA, ETATS-UNIS (20/07/1998) 1998 , vol. 3455, pp. 128-139.

13. Y. Zorian, A Distributed BIST Control Scheme for Complex VLSI Devices // Proc. 11th IEEE VLSI Test Symposium, April 1993.- pp.4-9.
14. S. Wang, S. Gupta, ATPG for Heat Dissipation Minimization During Test Application // Proc. IEEE International Conference, October 1994.- pp.250-258.
15. V. Dabholkar, S. Chakravarty, Two Techniques to Minimize Power Dissipation During Test Application in Scan Circuits // IEEE Asian Test Conference, November 1994.- pp.324-329.
16. A. Shen, A. Ghosh, S. Devadas, K. Keutzer An Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks // Proc. IEEE International Conference on Computer-Aided Design.- pp.402-407.
17. Brgles F., Bryan D., Kozminski K. Combinational profiles of sequential benchmark circuits // International symposium of circuits and systems, ISCAS-89.– 1989.– P.1929-1934.