



Получили систему из $n + 1$ линейных алгебраических уравнений с $n + 1$ неизвестными

$$\begin{cases} \sum_{k=0}^n a_k H^T A^k H = H^T A^0 P, \\ \sum_{k=0}^n a_k H^T A^{k+1} H = H^T A^1 P, \\ \dots \\ \sum_{k=0}^n a_k H^T A^{k+n} H = H^T A^n P. \end{cases}$$

Из линейной независимости строк следует, что детерминант не обращается в ноль, и решение системы единственное.

Легко убедиться, что уравнения (3) определяют минимум, так как матрица вторых производных (матрица Гессе) положительно полуопределена [4].

При выборе степени аппроксимирующего полинома повышение его степени целесообразно до такого значения, при котором отклонение измеренных температур от расчетных по модели остается больше погрешности измерений. В противном случае (при более высоких степенях) погрешность измерений прямо редуцируется в погрешность определения распределенного параметра по методу МНК. И метод теряет свои фильтрующие и стабилизирующие свойства.

Библиографический список

1. Ткаченко В.Н., Иванова А.А. Анализ температурных полей криволинейной МНЛЗ на основе математического моделирования // Материалы 3-й междунар. науч.-практ. конф. «Прогресивні технології у металургії сталі: XXI сторіччя». Донецк: ДонНТУ, 2007. С. 242–249.
2. Мацевитый Ю.М. Обратные задачи теплопроводности. В 2 т. Т.2. Приложения. Киев: Наук. думка, 2003.
3. Самарский А.А., Вабищевич П.Н. Численные методы решения обратных задач математической физики. М.: Едиториал УРСС, 2004.
4. Гантмахер Ф.Р. Теория матриц. М.: Наука, 1966. 576 с.
5. Иванова А.А. Математическая модель процесса затвердевания непрерывного слитка в зоне вторичного охлаждения. // Труды ИПММ НАН Украины. Донецк: ИПММ, 2006. Вып.12. С. 76-84.
6. Ткаченко В.Н. Моделирование тепловых процессов в автоматизированных системах обработки информации // Вісник Донецького національного університету. Сер. А. Природничі науки. 2002. №2. С. 379–383.

УДК 519.713:681.3

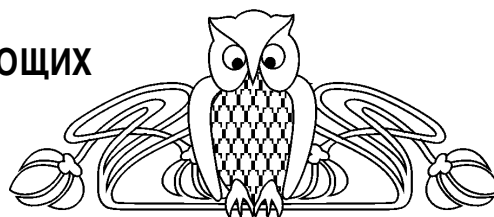
СИНТЕЗ ПСЕВДОСЛУЧАЙНЫХ КОНТРОЛИРУЮЩИХ ТЕСТОВ ДЛЯ ДИСКРЕТНОГО УСТРОЙСТВА

И.В. Уколов

Саратовский государственный университет,
кафедра математической кибернетики и компьютерных наук
E-mail: ukivan@yandex.ru

В статье рассматривается псевдослучайный метод построения контролирующих тестов для дискретных устройств, применимый как к комбинационным, так и к последовательностным устройствам. Используется энтропийный подход для отыскания оптимального распределения вероятностей входных наборов, что позволяет сокращать среднюю длину генерируемых тестов. Для определения упомянутых вероятностей решается многомерная задача поиска максимума выходной энтропии с применением генетического алгоритма. Рассмотрена задача моделирования исправного устройства и неисправных модификаций с использованием системы моделирования Active-HDL. Приведены результаты построения тестов для схем из международного каталога ISCAS'89. Выполнено их сравнение с данными, полученными другими авторами с использованием иных генетических алгоритмов.

© И.В. Уколов, 2008



Synthesis of Pseudorandom Test Patterns for the Discrete Device

I.V. Ukolov

In article the pseudorandom method of test patterns generation for discrete devices applicable for both to combinational devices and to sequential devices is considered. The entropy approach for search of optimum distribution of probabilities of input vectors is used that allows reducing average length of generated tests. For definition of the mentioned probabilities the multivariate problem of search of a maximum of output entropy with application of genetic algorithm is solved. The problem of simulation of the correct device and faulty devices is considered based on simulation system Active-HDL. Results of tests generation for circuits from international catalogue ISCAS'89 are produced. Their comparison with the data collected by other authors with use of other genetic algorithms is done.



ВВЕДЕНИЕ

В течение последних нескольких десятилетий произошло бурное развитие цифровой техники на сверхбольших интегральных схемах (СБИС). Согласно эмпирическому правилу, число транзисторов на кристалле СБИС для процессорных схем удваивается каждые 18 месяцев. Возрастание сложности СБИС, обусловленное развитием технологий производства интегральных схем, позволяет иметь в аппаратуре все большее число компонентов и реализовывать все более сложные функции. Наряду с этим происходит эволюция систем автоматизированного проектирования (САПР). Как альтернатива рисованию детализированных схем из низкоуровневых элементов появились языки описания аппаратуры высокого уровня (HDL). Используя современную САПР, один конструктор может создать сложный цифровой проект, для которого раньше бы потребовалось около десятка конструкторов. Все это создает благоприятные условия для разработки больших схем.

Однако увеличение размерности и сложности дискретных устройств (ДУ) влечет за собой усложнение проблемы тестирования и контроля этих устройств. Увеличение размерности и сложности схем на чипах, часто с сохранением количества внешних контактов, создает сложности при тестировании. Необходимо контролировать значительно больше логических элементов, используя то же самое количество контактов ввода-вывода. Как результат, затраты на тестирование составляют значительную долю стоимости производства. Поэтому появляется необходимость в создании новых методов тестирования или модификации методов, разработанных ранее.

Задача построения тестов является наиболее сложной для последовательностных логических схем, поскольку необходимо принимать во внимание начальные состояния элементов памяти, которые могут быть неизвестны, и явления состязания сигналов, вызванные временными задержками распространения сигналов.

1. ПОСТАНОВКА ЗАДАЧИ

В общем случае тест — это эксперимент, позволяющий подтвердить или опровергнуть корректность работы ДУ. Далее под контролирующими тестами понимаются последовательности входных воздействий, анализ выходных реакций на которые позволяет проверить исправность испытуемого ДУ. Под диагностическими тестами понимаются последовательности входных воздействий, позволяющие определить место расположения неисправностей, присутствующих в нем.

Пусть $A = \{A_1, A_2, \dots, A_f\}$ — конечное множество неисправных модификаций исправного устройства A_0 . Условимся, что функционирование A_0 отлично от функционирования любого $A_i \in A$.

Входную последовательность, выходные реакции на которую устройств A_0 и A_i ($i = \overline{1, f}$) различны, назовем проверяющей данную неисправность. Будем называть тестом, проверяющим неисправности множества A , входную последовательность, которая является проверяющей для всех $A_i \in A$. Тест называется полным, если он проверяет все неисправности из заданного класса.

При решении задач анализа и диагностирования ДУ наиболее широко используется класс логических неисправностей, при наличии которых связи в логической сети не нарушаются, а могут изменяться только функции логических элементов. К таким неисправностям относятся дефекты типа короткого замыкания и перепутывания соединений схемы. Часто неисправности такого рода могут быть описаны константными неисправностями, поскольку для большинства логических элементов обрыв входа эквивалентен подаче на этот вход постоянного сигнала 0 или 1. По числу одновременно присутствующих в схеме логических неисправностей различают одиночные, эквивалентные изменению функции только одного логического элемента (константные неисправности на одной линии схемы), и кратные неисправности, которые являются произвольным сочетанием одиночных неисправностей.

В статье рассматривается задача построения контролирующего теста наименьшей длины, обеспечивающего заданную полноту обнаружения неисправностей. Класс рассматриваемых неисправностей составляют одиночные константные неисправности на всех входах и выходах составляющих ДУ элементов. Под полнотой обнаружения неисправностей понимается процент обнаруженных неисправностей ДУ по отношению ко всему множеству его неисправностей.



2. МЕТОД РЕШЕНИЯ

Ниже рассматривается вероятностный подход для решения задачи, который позволяет избежать анализа структуры тестируемого ДУ, но требует моделирования исправного устройства.

Основная проблема, возникающая при генерации псевдослучайного теста для достаточно сложных устройств, — его большая длина. В [1] был предложен один из возможных путей решения этой проблемы на основе применения центрального понятия теории информации — энтропии. Там же показано, что максимизация выходной энтропии ДУ позволяет минимизировать длину псевдослучайного теста.

Следовательно, способ сокращения длины псевдослучайного теста при вероятностном тестировании дискретных устройств состоит в выборе оптимального распределения вероятностей входных наборов. В данной статье решается многомерная задача поиска максимума выходной энтропии ДУ в предположении, что вероятности на его входах различны. Ниже исследуется возможность ее решения с применением генетического алгоритма. Это позволяет получить оптимальное или близкое к оптимальному распределение вероятностей входных наборов ДУ, а значит, и сократить длину псевдослучайного теста.

Рассмотрим некоторое ДУ с n первичными входами и m первичными выходами. Пусть первичные входы ДУ являются независимыми и имеют различные вероятности u_i поступления сигнала «1» ($i = \overline{1, n}$). Совокупность таких вероятностей $\bar{u} = (u_1, u_2, \dots, u_n)$ назовем вектором входных вероятностей. Пусть все 2^m попарно различных двоичных кортежей длины m , представляющие все возможные выходные реакции рассматриваемого ЦУ, расположены и пронумерованы в лексикографическом порядке $(0, 0, \dots, 0), (0, 0, \dots, 1), \dots, (1, 1, \dots, 1)$. Обозначим $p_j(\bar{u})$ — вероятность появления на выходах ДУ кортежа с номером j ($j = \overline{1, 2^m}$) и назовем ее выходной вероятностью. Тогда выходная энтропия может быть определена выражением:

$$H_0(\bar{u}) = - \sum_{j=1}^{2^m} p_j(\bar{u}) \log p_j(\bar{u}).$$

В [1] получено соотношение между вероятностью $P(T)$ обнаружения всех неисправностей на тесте T длины L и выходной энтропией $H_0(\bar{u})$:

$$P(T) = 1 - \frac{1}{2^{H_0 L/k}},$$

где k — коэффициент, определяемый структурой логической схемы ДУ.

Из этого соотношения следует, что максимизация выходной энтропии ДУ минимизирует число тестовых наборов, необходимых для выявления всех неисправностей. Кроме того, вектор входных вероятностей \bar{u}_0 , при котором достигается максимальное значение энтропии $H_0(\bar{u})$, обеспечивает наибольшее значение величины $P(T)$.

Выходные вероятности являются неизвестными величинами и получение их в явном виде даже для сравнительно простых ДУ является задачей, сравнимой по сложности с построением детерминированного теста.

Поэтому предлагается заменить их соответствующими частотами $p_j^*(\bar{u})$ ($j = \overline{1, 2^m}$), вычисленными на некоторой случайной выборке достаточной длины из генеральной совокупности всех входных наборов мощности 2^n .

Итак, рассматривается следующая экстремальная задача: требуется найти точку $\bar{u}^* = (u_1^*, u_2^*, \dots, u_n^*)$, лежащую в области допустимых значений $\bar{u}^* \in U^n = \{\bar{u} = (u_1, u_2, \dots, u_n) \mid 0 \leq u_i \leq 1\}$ ($i = \overline{1, n}$), в которой функция

$$H_0^*(\bar{u}) = - \sum_{j=1}^{2^m} p_j^*(\bar{u}) \log_2 p_j^*(\bar{u}) \quad (1)$$

имеет максимальное значение.

Особенность этой задачи заключается в том, что функция $H_0^*(\bar{u})$ является случайной и ее явный вид, благодаря замене выходных вероятностей частотами, неизвестен. Поэтому непосредственное вычисление значений функции $H_0^*(\bar{u})$ в различных точках $\bar{u} \in U^n$, а тем более ее производных,



невозможно. Однако для всякого $\bar{u} \in U^n$ с использованием результатов моделирования ДУ на тесте с заданным распределением вероятностей на входах можно получить значения $p_j^*(\bar{u})$ ($j = \overline{1, 2^m}$) и вычислить $H_0^*(\bar{u})$ по формуле (1).

Поиск значений входных вероятностей, максимизирующих функцию (1), можно осуществить разными способами. Можно воспользоваться любым известным методом поиска экстремума. Как уже отмечалось выше, в данной работе для этого использовался генетический алгоритм.

3. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Основные принципы генетических алгоритмов сформулировал John Holland [2]. Генетические алгоритмы (ГА) работают с совокупностью особей — популяцией, каждая из которых представляет возможное решение проблемы. Каждая особь оценивается фитнес-функцией, показывающей, насколько хорошо данная особь соответствует решению задачи. Наиболее приспособленные особи, т.е. особи, имеющие наибольшее значение фитнес-функции, получают возможность перейти в следующее поколение. Это приводит к появлению новых особей, которые сочетают в себе некоторые характеристики, наследуемые ими от родителей. Таким образом, исследуются наиболее перспективные участки пространства поиска. Наименее приспособленные особи с меньшей вероятностью смогут воспроизвести потомков, так что те свойства, которыми они обладали, будут постепенно исчезать из популяции в процессе эволюции.

Генетический алгоритм получает на вход начальную популяцию, равномерно покрывающую пространство поиска. Работа ГА представляет собой итерационный процесс, который продолжается до тех пор, пока не выполнится заданное число итераций. На каждой итерации производится построение следующего поколения особей — новой популяции. После завершения работы алгоритма решением будет являться особь с наибольшим значением фитнес-функции в последнем поколении.

В контексте данной статьи особи — это вектора входных вероятностей $\bar{u} = (u_1, u_2, \dots, u_n)$, описанные выше. Критерием отбора в следующее поколение или фитнес-функцией является выходная энтропия, вычисляемая по формуле (1).

При построении следующей популяции из исходной последовательно выполняются основные операторы ГА — оператор селекции, кроссовера, мутации и редукции.

Оператор селекции производит пропорциональный отбор особей из исходной популяции. Реализация данного типа селекции основана на k запусках рулетки, содержащей по одному сектору для каждого члена популяции размером, пропорциональным величине фитнес-функции данной особи, где k — это количество особей в исходной популяции.

Отобранные k особей подвергаются равномерному кроссоверу. Особи случайным образом разбиваются на $k/2$ штук пар. Для каждой пары с заданной вероятностью может применяться оператор кроссовера. Равномерный кроссовер работает следующим образом. Производится случайная генерация бинарного вектора \bar{v} длины n . Первый потомок получает сегменты первого родителя, соответствующие единичным разрядам вектора \bar{v} , и сегменты второго родителя, соответствующие нулевым разрядам вектора \bar{v} . Второй потомок строится аналогично — используются сегменты первого родителя, соответствующие нулевым разрядам \bar{v} , и сегменты второго родителя, соответствующие единичным разрядам \bar{v} .

К популяции, полученной на предыдущем этапе, с заданной вероятностью применяется оператор мутации. При этом производится изменение случайно выбранного гена особи.

Оператор редукции выполняет сокращение количества полученных особей до размера исходной популяции. В соответствии с ним в новую популяцию попадают k особей, имеющих наибольшее значение фитнес-функции — элитный тип оператора редукции.

Оценка эффективности программной реализации данного ГА приведена ниже.

4. ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

В технической диагностике ДУ широко применяется логическое моделирование. Логическое моделирование включает в себя построение модели ДУ — системы отношений, описывающей поведение исследуемого устройства с заданной точностью, и дальнейший анализ поведения этой модели на за-



данной последовательности входных воздействий. В данной статье логическое моделирование используется для вычисления фитнес-функции при оптимизации входных вероятностей с использованием ГА по формуле (1) во время построения контролирующего теста, а также для определения полноты полученного теста, получения списка проверенных и не проверенных неисправностей и другой диагностической информации.

На сегодняшний день очень широкое распространение получили языки описания аппаратуры. Наиболее распространенным языком этого класса, специфицированным международными стандартами, является язык VHDL [3], который разработан в рамках проекта создания нового поколения высокоскоростной элементной базы (VHSIC).

5. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

В рамках программного проекта DiagnosticSuite были разработаны и реализованы алгоритмы, описанные выше, и система логического моделирования ДУ с неисправностями, использующая для моделирования исправных и неисправных устройств среду проектирования и моделирования ДУ Active-HDL [<http://www.aldec.com/>]. Проект реализован на платформе Microsoft .NET framework [<http://www.microsoft.com/>].

Исходными данными для моделирования являются текстовые описания схем из международного каталога ISCAS'89, на которых принято производить тестирование новых методов моделирования и построения тестов.

Взаимодействие системы моделирования и Active-HDL реализовано через перенаправление потоков ввода/вывода в асинхронном режиме автономного инструмента моделирования VSimSA, спроектированного и оптимизированного для продолжительного моделирования и пакетной обработки, входящего в среду Active-HDL. Таким образом, все возможности, предоставляемые Active-HDL, как, например, просмотр временных диаграмм, списка значений сигналов, запись истории значений сигналов в базу данных asdb и др., остаются доступными.

Моделирование может проходить в двух режимах: последовательном и параллельном. Во втором случае поддерживается пул процессов моделирования VSimSA с очередью задач на выполнение.

При инициализации сессии моделирования каждого процесса VsimSA производится трансляция текстового описания ДУ в структурное описание схемы на языке VHDL. Функциональное описание базисных элементов собрано в отдельную библиотеку. При этом используется тип сигналов в схеме STD_LOGIC, что означает моделирование в 9-значном алфавите. Для синхронизации триггеров в схеме используется дополнительный первичный вход CLK. В описание ДУ на VHDL производится добавление служебного VHDL процесса, который собирает всю необходимую диагностическую информацию во время моделирования и по запросу выдает ее на стандартный вывод. Для подачи сигналов на первичные входы моделируемого устройства используется VSimSA команда «force <signal> <value> [<time>] [, <value> <time>]».

В каталоге схем ISCAS'89 также содержатся данные о множествах проверяемых неисправностей, которые используются при моделировании ДУ с неисправностями. Моделирование с неисправностями производится в параллельном режиме. Для каждой следующей неисправной модификации устройства в списке осуществляется переинициализация сессии моделирования командой VSimSA «restart» и производится вставка неисправности на требуемый вход или выход элемента схемы с помощью ключа «-freeze» команды «force».

6. ОЦЕНКА ЭФФЕКТИВНОСТИ

В этом разделе приведены результаты работы программы синтеза псевдослучайных контролируемых тестов для восьми схем s298, S386, S641, S713, S1196, S1238, S1488, S1494 из ISCAS'89 каталога.



Основные характеристики используемых схем приведены в табл. 1.

Таблица 1

Описания используемых схем

Схема	Описание
S298	3 первичных входа, 6 первичных выходов, 14 D-триггеров, 44 инвертера, 75 вентиляей
S386	7 первичных входа, 7 первичных выходов, 6 D-триггеров, 41 инвертера, 118 вентиляей
S641	35 первичных входа, 24 первичных выходов, 19 D-триггеров, 272 инвертера, 107 вентиляей
S713	35 первичных входа, 23 первичных выходов, 19 D-триггеров, 254 инвертера, 139 вентиляей
S1196	14 первичных входа, 14 первичных выходов, 18 D-триггеров, 141 инвертера, 388 вентиляей
S1238	14 первичных входа, 14 первичных выходов, 18 D-триггеров, 80 инвертера, 428 вентиляей
S1488	8 первичных входа, 19 первичных выходов, 6 D-триггеров, 103 инвертера, 550 вентиляей
S1494	8 первичных входа, 19 первичных выходов, 6 D-триггеров, 89 инвертера, 558 вентиляей

Результаты работы программы для названных схем приведены в табл. 2. Для каждой схемы указаны общее количество неисправностей, количество проверенных неисправностей, полнота и длина построенных тестов.

Таблица 2

Результаты численных экспериментов

Схема	Всего/Проверено	Полнота, %	Длина
S298	308/265	86,04	382
S386	384/234	60,94	374
S641	467/396	85,16	808
S713	581/467	80,38	428
S1196	1242/907	73,03	1637
S1238	1355/950	70,11	1118
S1488	1486/1433	96,43	1953
S1494	1506/1439	95,55	1699

В табл. 3 приведены результаты в сравнении с результатами, полученными другими авторами [4], и с результатами генерации теста с вероятностями 0,5 по всем входам, что обычно используется в качестве стандартного распределения.

Таблица 3

Сравнение результатов

Схема	Данные, полученные с помощью предложенного метода		Данные, полученные с помощью ГА из [4]		Данные по синтезу тестов со стандартным распределением	
	Полнота, %	Длина	Полнота, %	Длина	Полнота, %	Длина
S298	86,04	382	82,79	637	48,21	382
S386	60,94	374	68,49	606	55,47	875
S641	85,16	808	86,30	727	83,66	997
S713	80,38	428	80,38	677	75,90	641
S1196	73,03	1637	96,38	1642	87,44	1978
S1238	70,11	1118	90,18	1138	83,39	1982
S1488	96,43	1953	70,46	1975	58,81	1962
S1494	95,55	1699	70,58	1712	57,47	1843



Анализ экспериментальных данных, полученных для приведенных выше и других схем, и их сравнение с результатами, опубликованными в [4], говорит о достаточно высокой эффективности предложенного алгоритма синтеза случайных тестов.

Основным достоинством приведенного алгоритма является необходимость моделирования только исправного ДУ при достижении высокого уровня покрытия тестируемых схем тестовыми наборами. Моделирование с неисправностями используется исключительно для получения диагностической информации о построенном тесте.

Библиографический список

1. Agrawal V.D. An Information Theoretic Approach to Digital Fault Testing // IEEE Transactions on Computers. 1981. V. 30. P. 582–587.
2. Holland J.H. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.
3. Суворова Е.А., Шейнин Ю.Е. Проектирование цифровых систем на VHDL. СПб.: БХВ-Петербург, 2003.
4. Скобцов Ю.А., Скобцов В.Ю. Логическое моделирование и тестирование цифровых устройств. Донецк: Изд-во Донецк. техн. ун-та, 2005.

УДК 519.713:681.3

СИНТЕЗ КОНТРОЛИРУЮЩИХ ТЕСТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ С ПРИМЕНЕНИЕМ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

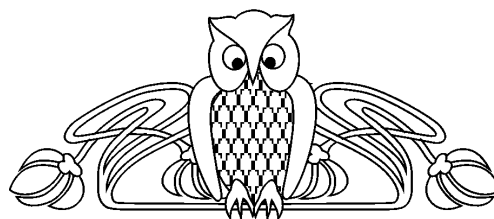
Е.В. Уколова

Саратовский государственный университет,
кафедра математической кибернетики и компьютерных наук
E-mail: b4456@yandex.ru

В статье предложен метод построения контролирующих тестов для дискретных устройств, основанный на генетическом алгоритме. Работоспособность и эффективность метода проверялись путем построения тестов для схем, приведенных в каталоге ISCAS'89. Для моделирования работы схем и генерации тестов была написана программа на C++, Visual Studio 2005. Неисправности эмулируются программной установкой соответствующего объекта в памяти в особое состояние, при котором он выполняет функцию неисправного элемента. Генерация тестов осуществлялась при различных значениях и конфигурациях параметров генетического алгоритма: изменялись вид селекции, мощность популяции, вероятность мутации, процент элитных особей, максимальное количество итераций. В статье приведены результаты построения тестов, а также сравнение с данными других авторов.

ВВЕДЕНИЕ

В современном мире бурно развиваются технологии производства, функции дискретных устройств становятся все более разнообразными, а их технические характеристики непрерывно улучшаются. В связи с этим происходит постоянное усложнение элементной базы дискретных устройств. Как следствие, растет вероятность появления неисправностей и сложность их обнаружения. Таким образом, весьма актуальной задачей является разработка методов обнаружения неисправностей. Один из таких методов заключается в подаче на устройство теста — специальной последовательности, особенность которой заключается в том, что реакции на нее исправного и неисправного устройств различаются. В качестве входных используются либо рабочие воздействия (то есть такие, которые поступают на устройство в процессе его функционирования по назначению), либо специально генерируемые тестовые воздействия. Сложность современных дискретных устройств приводит к необходимости автоматизации процесса построения тестов. Одним из способов достижения этого является применение генетического алгоритма (ГА).



The Construction of Supervisory Test with the Use of the Genetic Algorithm

E.V. Ukolova

The article describes a control test generation method for discrete devices based on the genetic algorithm. The method operability and effectiveness have been checked by means of creating tests for circuits listed in the ISCAS' 89 catalogue. The C++ (Visual Studio 2005) program has been implemented in order to simulate circuit and generate tests. Faults are simulated by programmatically setting an appropriate object in storage to a special state, in which it acts as a faulty component. Test generation has been executed using different values and configurations of the genetic algorithm parameters: selection, population capacity, mutation probability, percent of elite individuals, maximal amount of iterations were changed. The final results and comparison with results of other authors are included into the report.