

АЛГОРИТМ ДИАГНОСТИЧЕСКОГО МОДЕЛИРОВАНИЯ СБИС

УДК 681.518:681.326.7

ИВАНОВ Дмитрий Евгеньевич

к.т.н., доцент, старший научный сотрудник института прикладной математики и механики НАН Украины.

Научные интересы: техническая диагностика, моделирование и тестирование цифровых устройств, генетические алгоритмы, эволюционные вычисления, параллельные алгоритмы.

e-mail: ivanov@iamm.ac.donetsk.ua

ВВЕДЕНИЕ

Диагностические алгоритмы моделирования относятся к алгоритмам моделирования с неисправностями. Однако, если обычно моделирование с неисправностями служит для определения полноты проверочных тестов (отношение числа проверенных неисправностей к общему числу рассматриваемых неисправностей), то диагностическое моделирование используется для определения свойства последовательностей относительно некоторых диагностических мер. Алгоритмы диагностического моделирования также являются важным компонентом систем генерации диагностических тестов.

Алгоритмы диагностического моделирования рассматриваются исследователями гораздо реже в сравнении с алгоритмами определения полноты тестов. Это связано с тем, что задача построения диагностических последовательностей имеет существенно большую сложность, чем задача построения проверяющих тестов. Проверяющие тесты определяют для каждой неисправности являются ли выходные реакции цифрового устройства (ЦУ) в её присутствии аналогичными исправному устройству. В противоположность им диагностический тест должен отличить поведение некоторого неисправного ЦУ не только от исправного, но и от всех других неисправных ЦУ. Сложность задачи построения диагностических последовательностей обуславливается применяемыми алгоритмами, которые в основе своей разработаны для автоматного уровня представления

ЦУ [1-2], а впоследствии распространены на структурный.

В последнее десятилетие распространение получили методы построения идентифицирующих последовательностей (ИдП), которые основаны на моделировании. К ним относятся и эволюционные алгоритмы (ЭА) построения ИдП, в частности генетические алгоритмы (ГА) [3-4], либо алгоритмы симуляции отжига (СО) [5]. Особенностью таких алгоритмов является возможность работы с ЦУ достаточно большой размерности, которые на практике представлены сверхбольшими интегральными схемами (СБИС). Первоначально эволюционные алгоритмы разрабатывались для генерации проверяющих тестов, а позднее и для построения диагностических последовательностей. В частности в [6-8] описываются ГА построения диагностических тестов. В данных алгоритмах вычисление оценки потенциальных решений производится на основании диагностического моделирования ЦУ при подаче на его вход особи-последовательности. Данный алгоритм получает на вход некоторый класс неразличимых неисправностей. Его задача путём моделирования заданной последовательности на неисправностях данного класса определить: распадается ли данный класс на несколько других классов неразличимых неисправностей? При этом, если такого разбиения не произошло, алгоритм моделирования должен возратить меру того, что заданная последовательность способна произвести такое разбиение на подклассы.

При построении автоматизированной системы генерации диагностических тестов [9], которая будет включать и соответствующий ЭА, важными являются два режима работы алгоритма диагностического моделирования. Первый режим используется итеративно для вычисления заданных диагностических мер для построенной входной последовательности. В случае, если не достигнут показатель качества, следует продолжить генерацию таких последовательностей. Второй режим используется в эволюционном (в частности, основанном на ГА) алгоритме генерации диагностических тестов, а его применение в данном контексте описано выше.

Цель данной статьи разработка метода диагностического моделирования, который предназначен для работы в соответствующей подсистеме системы моделирования и диагностики ASMID-Evolution [10], и который позволяет выполнять оценки качества заданной входной последовательности для двух указанных выше режимов работы.

ДИАГНОСТИЧЕСКИЕ МЕРЫ

В данной работе в качестве модели СБИС используются синхронные цифровые последовательностные устройства, заданные на логическом уровне [11]. В таких ЦУ выделяют комбинационную часть и элементы состояний, реализованные на основе D-триггеров. Моделирование поведения ЦУ осуществляется итеративным моделированием комбинационного эквивалента. В качестве моделей неисправностей мы рассматриваем одиночные константные неисправности «константа 0» и «константа 1». Однако без потери общности при реализации предлагаемый метод может быть распространён на другие типы неисправностей, для которых разработан соответствующий метод моделирования.

Дадим необходимые определения.

Определение 1. Пусть задано исправное ЦУ A_0 и класс неисправностей $F = \{f_1, \dots, f_n\}$, который соответственно порождает класс неисправных устройств $A = \{A_1, \dots, A_n\}$. Входная последовательность T , которая может отличить поведение произвольного устройства A_i от поведения исправного A_0 , а также поведения всех остальных неисправных устройств, называется диагностической.

Определение 2. Для заданного ЦУ A и двух неисправностей f_1 и f_2 последовательность T называется различающей, если выходные реакции $A_1(T)$ и $A_2(T)$ различны хотя бы для одного входного вектора.

Определение 3. Все неисправности f_1, \dots, f_n , которые не различаются заданной входной последовательностью T для заданного ЦУ A , принадлежат к одному классу неразличимости F' относительно последовательности T .

Определение 4. Для заданного ЦУ A и заданного класса неисправностей $F = \{f_1, \dots, f_n\}$ последовательность T называется различающей, если существует хотя бы одна неисправность $f_j \in F$ такая, что $A_j(T) \neq A_i(T)$ (для $i = \overline{1, n}$ кроме $i = j$) хотя бы для одного входного вектора.

В соответствии с определением 4, если для некоторой неисправности $f_j \in F$ было обнаружено, что $A_j(T) \neq A_i(T)$ (для $i = \overline{1, n}$ кроме $i = j$), то она сразу будет образовывать новый класс неразличимых неисправностей. Однако выделение такого класса с одной неисправностью $F' = \{f_j\}$ в первый момент времени, когда обнаружено выполнение свойства определения 4, является неверным, поскольку при дальнейшем моделировании возможно выполнение данного условия для других неисправностей из F .

При построении алгоритма диагностического моделирования особое внимание следует уделить пониманию принципов построения диагностических мер. Рассмотрим наиболее распространенные из них.

Для заданной входной последовательности $T = T_1 T_2 \dots T_m$ и множества неисправностей $F = \{f_1, \dots, f_n\}$ диагностические свойства могут быть представлены в виде матрицы выходных реакций Y . Данная матрица будет иметь размерность $(n + 1) * m$:

$$Y = \begin{pmatrix} y_{01} & \dots & y_{0m} \\ y_{11} & & y_{1m} \\ \dots & & \dots \\ y_{n1} & & y_{nm} \end{pmatrix}, \quad (1)$$

где элементы первой строки y_{0j} показывают реакции исправного ЦУ A_0 после моделирования набора с индексом j , а все последующие – аналогичные ре-

акции ЦУ в присутствии соответствующих неисправностей. Поскольку обычно ЦУ имеют более одного выходного сигнала, то элемент матрицы y_{ij} является двоичным вектором с размерностью k , показывающей число внешних выходов ЦУ.

Наблюдение поведения ЦУ только по выходным значениям характерно для их автоматного задания. При обработке ЦУ на структурном уровне предполагается, что разработчику доступны аналогичные сведения об элементах состояний ЦУ, а также для множества контрольных точек, узлов и т.п. В этом случае диагностическая матрица вида (1) может быть расширена естественным образом за счёт увеличения разрядности элементов y_{ij} .

Для реально проектируемых ЦУ матрица выходных реакций в виде (1) явно не задаётся ввиду большой размерности, что требует для её хранения чрезмерных затрат памяти. Расширение же хранимой диагностической информации на множество узлов и элементов состояний ЦУ будет далее ухудшать ситуацию. Поэтому исследования ряда авторов направлены на сжатие матриц выходных сигналов и построение сжатых диагностических словарей [12-16].

Другой подход используется в [17-18]. Для обеспечения диагностической информации по результатам моделирования строится матрица различимости:

$$D = \begin{pmatrix} d_{11} & \dots & d_{1m} \\ \dots & \dots & \dots \\ d_{n1} & \dots & d_{nm} \end{pmatrix}. \quad (2)$$

Данная матрица имеет размерность $n * n$ (где n - число неисправностей в полном списке F). Элемент матрицы d_{ij} показывает:

$$d_{ij} = \begin{cases} 0, & \text{если выходные реакции} \\ & \text{А в присутствии неисправностей} \\ & \text{с номерами } i \text{ и } j \text{ одинаковы;} \\ 1, & \text{в противном случае.} \end{cases} \quad (3)$$

Оба типа рассматриваемых матриц используются для вычисления диагностических мер заданных последовательностей. Рассмотрим основные используемые на практике диагностические меры.

Диагностическая мера, называемая диагностической разрешимостью, предложена в [19].

Определение 5. Диагностической разрешимостью (ДР) данной последовательности S относительно данного множества неисправностей F для заданного ЦУ A называется отношение числа различных пар неисправностей в общему числу пар неисправностей:

$$ДР = \frac{\text{число_различимых_пар_неисправностей}}{\text{общее_число_неисправностей}}. \quad (4)$$

Определение диагностической меры относительно пар различных неисправностей ведёт к тому, что ряд алгоритмов диагностического моделирования основаны не на заполнении диагностической матрицы, а на моделировании пар всевозможных неисправностей из исходного множества F_0 . Пример такого алгоритма разработан в [20].

Определение 6. Диагностической мощностью (ДМ) данной последовательности S относительно данного множества неисправностей F для заданного ЦУ A называется отношение числа полностью отличимых неисправностей к общему числу неисправностей:

$$ДМ = \frac{\text{число_различимых_неисправностей}}{\text{общее_число_неисправностей}}. \quad (5)$$

Определение 7. Диагностическим покрытием (ДП) данной последовательности S относительно данного множества неисправностей F для заданного ЦУ A называется отношение числа групп неразличимых неисправностей к общему числу неисправностей:

$$ДП = \frac{\text{число_групп_неразличимых_неисправностей}}{\text{общее_число_неисправностей}}. \quad (6)$$

Методы вычисления указанных диагностических мер с применением матрицы различимостей разрабатываются, например, в [21]. Временная сложность таких алгоритмов $O(n^2)$, что также с учётом размерности проектируемых ЦУ является достаточно большой. Для ускорения работы применяются различные подходы. Например, в [22] используются оптимизационные эвристические процедуры основанные на понятии z -диагностирования, тогда как в [23] используется аппаратный параллелизм.

В [24] предложена мера, которая определяет множества эквивалентных неисправностей относительно заданной тестовой последовательности S , т.е. показывает число классов неразличимых неисправностей мощности 1, 2 и т.д. В соответствии с данной мерой диагностические качества последовательности S относительно множества неисправностей F для заданного

ЦУ A показывает последовательность чисел $\{h_i\}$, в которой i -й элемент показывает число классов неразличимости неисправностей мощности i . В такой интерпретации для диагностической последовательности S в идеальном случае (определение 1) длина последовательности чисел $\{h_i\}$ будет равна 1, а элемент последовательности h_1 будет равен общему числу рассматриваемых неисправностей $n = |F|$, показывая, что поведение каждого неисправного ЦУ отличается от поведения всех остальных при моделировании на последовательности S .

В [17] эта мера ДП обобщена таким образом, чтобы учитывать значение неопределённого сигнала u из алфавита E_3 при моделировании поведения последовательностных ЦУ, которые начинают работу из полностью неопределённого состояния $Z = uu...u$. В том случае, если в присутствии одной неисправности f_i ЦУ порождает на выходе значение 0 из алфавита моделирования E_3 , а в присутствии другой неисправности f_j на том же выходном контакте наблюдается значение $u \in E_3$, то в этом случае нельзя определённо говорить о различимости неисправностей f_i и f_j . На самом деле данное свойство различимости будет зависеть от начального состояния: для некоторых начальных состояний такие неисправности будут различимы, а для некоторых – нет. Однако гарантировать различимости неисправностей f_i и f_j нельзя. Поэтому делается предположение, что выходные значения 0 и $1 \in Z_3$ неотличимы от значения $u \in Z_3$.

Очевидно, что методы вычисления диагностических метрик на основании заполнения матриц различимости являются достаточно медленными ввиду необходимости заполнения матриц большой размерности и дальнейшей их обработки.

В противоположность им предложены подходы [25], которые сразу по результатам моделирования текущего входного набора S_i обрабатывают списки неразличимых неисправностей без заполнения матриц различимости. Преимуществом таких подходов является то, что неисправности, которые образуют класс неразличимости мощности 1 сразу могут быть исключены из дальнейшего рассмотрения. Существенное улучшение

временных характеристик метода при этом достигается за счёт исключения моделирования и операций сравнения с такими неисправностями. Требования к емкостным ресурсам в таком методе существенно снижаются за счёт того, что не происходит явного заполнения матрицы выходных реакций ЦУ. Необходимая информация о выходных реакциях используется ситуативно при обработке списков.

ДИАГНОСТИЧЕСКАЯ МЕРА В ЭВОЛЮЦИОННЫХ АЛГОРИТМАХ ПОСТРОЕНИЯ ДИАГНОСТИЧЕСКИХ ТЕСТОВ

Рассмотренные диагностические меры вычисляются для получения оценки качества диагностических последовательностей. С другой стороны, диагностическое моделирование используется также при построении таких последовательности с целью оценки качества потенциальных решений. В том случае, когда на вход алгоритма диагностического моделирования поступает множество F' , которое не разбивается на меньшие заданной последовательностью S , такие меры ничего не скажут о качестве последовательности. Наиболее ценной информацией в данном случае было бы знать: насколько близко последовательность S подошла к тому, чтобы стать разбивающей для F' . Эта информация была бы очень полезной в ГА построения диагностических последовательностей. Для таких случаев в [6] вводится другая мера, которая также является оценочной функцией потенциального решения.

Сначала оценка вычисляется для каждого входного вектора, который входит в данную последовательность. Для класса неразличимости F_i и k -го вектора последовательности T_j данная функция имеет следующий вид:

$$E(T_j^k, F_i) = k_1 \cdot \sum_{n=1}^{N_{mp}} R_n^1(T_j^k, F_i) + k_2 \cdot \sum_{n=1}^{N_{вент}} R_n^2(T_j^k, F_i), \quad (7)$$

где:

- $R_n^1(T_j^k, F_i)$ – функция различия выходов триггеров;
- $R_n^2(T_j^k, F_i)$ – функция различия выходов вентиля;

- k_1 и k_2 – нормирующие коэффициенты, показывающие, что различия значений на выходах триггеров важнее различия на выходах элементов;
- $N_{\delta\delta}$ и $N_{\dot{\alpha}\dot{\alpha}}$ – число триггеров и выходов элементов в схеме соответственно.

Функции различия определяются следующим образом:

$$R_n^\alpha(T_j^k, F_i) = \begin{cases} 0, & \text{если выходы элементов} \\ & \text{с индексом } n, \text{ которые принадлежат} \\ & \text{классу } F_i, \text{ одинаковы} \\ & \text{после подачи набора } T_j^k; \\ 1, & \text{иначе;} \end{cases} \quad (8)$$

где $\alpha = 0,1$ для множества триггеров и выходов элементов схемы соответственно.

На основании оценки каждого вектора вычисляется оценочная функция всей входной последовательности, которая имеет вид:

$$E(T_j, F_i) = \max_k (E(T_j^k, F_i)). \quad (9)$$

Как видно из такого представления оценочной функции, диагностическая информация собирается не только по внешним выходам ЦУ, но и по множеству псевдовыходов. Смысл такого построения функции оценки в том, что если различия в поведении устройств на внешних выходах нет, то такое различие на псевдовыходах позволяет в дальнейшем распространить его на внешние выходы. Причём чем больше такое различие на псевдовыходах, тем скорее удастся распространить его на выходы ЦУ. Продолжая данную идею, можно в вид оценочной функции (1)-(3) добавить аналогичный член, который, например, будет соответствовать множеству контрольных точек, выбранных разрабатчиком.

ПРЕДЛАГАЕМЫЙ МЕТОД

Алгоритм диагностического моделирования может быть надстроен над алгоритмом моделирования ЦУ с неисправностями. Для этого в такой алгоритм необходимо добавить процедуры заполнения диагностической матрицы и вычисления диагностических метрик, либо обработчик списков неразличимых неисправностей.

Предлагаемый метод диагностического моделирования основывается на параллельном методе моделирования последовательностных ЦУ с неисправностями,

который описан в [26]. В предлагаемом методе в качестве техники для вычисления диагностических мер будем использовать списки множеств неразличимых неисправностей. Алгоритм реализует моделирование поведения ЦУ в трёхзначном алфавите E_3 .

В исходный алгоритм необходимо добавить процедуры работы со списками неисправностей. Пусть F' – текущий список, который содержит множества неразличимых неисправностей.

В первом режиме работы необходимо построение начального списка неразличимых неисправностей $F' = \{F_0\}$, где F_0 – полный список неисправностей ЦУ. Как и в исходном алгоритме моделирования с неисправностями данный список F_0 сжимается для исключения обработки эквивалентных неисправностей. Во втором режиме работы данный список F' передаётся из алгоритма построения диагностических тестов.

В обоих режимах работы после моделирования очередного входного набора на всех текущих списках неисправностей необходимо обновить списки неразличимых неисправностей F' . Это выполняется путём сравнения выходов неисправных ЦУ по каждому из множеств входящих в F' . Если такое различие обнаружено, то множество распадается на подмножества неразличимых неисправностей. Техника такого разбиения списков неисправностей показана на рис. 1. Здесь моделирование выполняется для начального списка $F' = \{F_0\} = \{\{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}\}$ и входной последовательности $T = T_1 \dots T_5$. Результатом являются 6 множеств, содержащих по одной неисправности: $F_3 = \{f_1\}$, $F_7 = \{f_2\}$, $F_8 = \{f_7\}$, $F_9 = \{f_8\}$, $F_{10} = \{f_3\}$, $F_{11} = \{f_6\}$, а также множество $F' = \{F_4\} = \{\{f_4, f_5\}\}$, содержащее в свою очередь одно множество F_4 мощности два.

Для второго режима работы в составе эволюционного алгоритма построения диагностических последовательностей возможно два случая. В первом случае для входного множества F' произошло разбиение на подмножества (как в примере выше). В этом случае на верхний уровень будет передан соответствующий признак, а также списки множеств мощности один и список F' , соответствующий моменту окончания моделиро-

вания. Такая ситуация характерна для начального этапа работы генератора диагностических тестов.

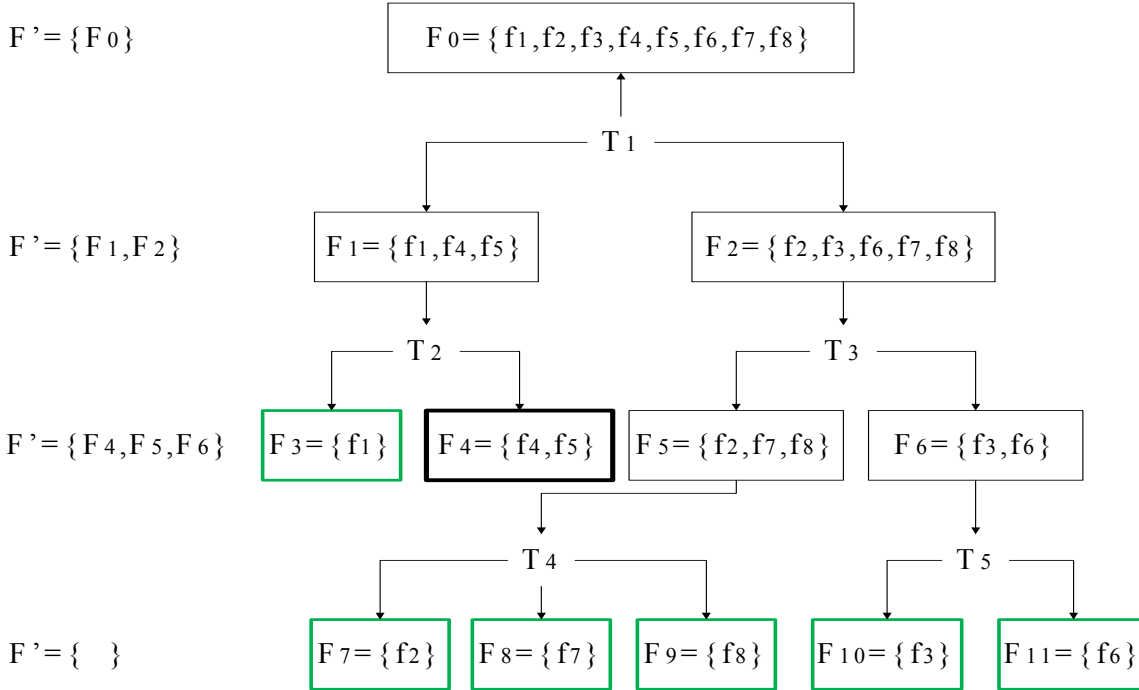


Рисунок 1 – Пример разбиения списка неразличимых неисправностей в процессе моделирования

Во втором случае разбиения F' не происходит. Тогда для F' по результатам моделирования вычисляется мера (9), которая передаётся в качестве результата на верхний уровень для принятия решения.

Также для первого режима работы после окончания моделирования на основании полученных списков неисправностей происходит вычисление метрик (4)-(6).

Алгоритмическая реализация описанного метода в виде псевдокода приведена ниже.

Алгоритм А1.

```

Диагностическое_моделирование ( A ,
F' , T )
{
    if ( режим == 1 )
    {
        F'
=ПостроениеСпискаНеисправностей ( ) ;
    }
    // цикл по входным наборам
    for ( int i=0 ; i<Длина ( T ) ;
i++ )
    {
        Набор=Ti ;
    }
}
    
```

```

SVI=МоделироватьИсправнуюСхему ( A ,
Набор ) ;
// цикл по множествам нераз-
личимых неисправностей
for ( int j=0 ; j<|F'| ; j++ )
{
    Fj = F'[j] ;
    while ( есть непроверенные
неисправности в Fj ) ;
    {
        Груп-
па=СформироватьГруппуНеисправностей (
СписокНеисправностей ) ;
        МоделироватьНеисправныеС-
хемыГруппы ( A , Набор , Группа ) ;
        СохранитьСостоянияДляНепроверенныхНе
исправностей ( SVI , Группа ) ;
    } // конец цикла по неис-
правностям в множестве
    ПроверитьРазличимость ( ) ;
    СформироватьНовыеМножества (
F' , Fj ) ;
}
    
```

```

УдалениеМножествМощностиОдин ( F' );
if( режим == 2 )
{
    ВычислениеПромежуточнойМеры();
}
} // конец цикла по множествам
неразличимых неисправностей
} // конец цикла по входным наборам
Мера=ВычислениеДиагностическихМер();
return ( F', Мера);
} // конец алгоритма диагностического моделирования

```

В данном алгоритме выполняется моделирование всех неразличимых групп неисправностей для каждого входного набора заданной последовательности T . Моделирование неисправных ЦУ из одного множества неразличимых неисправностей $F_j \in F'$ выполняется параллельно в разрядах машинного слова. Такой подход соответствует базовому алгоритму [26]. После того как выполнено моделирование всех неисправностей из множества F_j производится сравнение их выходных реакций в функции «ПроверитьРазличимость()». По результатам этого сравнения в функции «СформироватьНовыеМножества()» при необходимости модифицируется список множеств неразличимых неисправностей F' . Если в результате будут обнаружены множества мощности один, то в функции «УдалениеМножествМощностиОдин()» они будут исключены из F' , что предотвратит их дальнейшее рассмотрение.

Также для второго режима работы после моделирования очередного входного набора вычисляется промежуточная мера в виде (7).

Вычисление окончательных мер (4)-(6) или (9) в зависимости от режима работы выполняется после моделирования всей последовательности T в функции «ВычислениеДиагностическихМер()».

Результатом работы алгоритма является новый список F' множеств неразличимых неисправностей и вычисленная диагностическая мера (в зависимости от режима работы).

ЛИТЕРАТУРА:

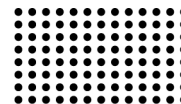
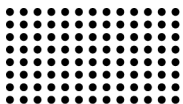
1. Гилл А. Введение в теорию конечных автоматов /А. Гилл. — М.: Наука, 1966. — 272 с.

В [27] предложен параллельный метод моделирования с неисправностями, основанный на алгоритме [26], который предназначен для работы на многоядерных рабочих станциях. Фактически, данный метод является параллельной версией базового алгоритма данной работы для многопроцессорных ВС с общей памятью. В данном методе выполняется параллельное моделирование нескольких групп неисправностей для текущего входного набора, причём каждая такая группа моделируется независимо на своём вычислительном ядре. Очевидно, что подобная техника параллельного моделирования также может быть распространена на предлагаемый метод диагностического моделирования. При этом на отдельном узле следует выполнять параллельное по разрядам моделирование неисправностей из одной группы неразличимых неисправностей F_j . Разработка такой параллельной версии метода является ближайшей задачей авторов.

ЗАКЛЮЧЕНИЕ

Разработан алгоритм диагностического моделирования, который для повышения гибкости использования поддерживает два режима работы. В первом режиме для заданной входной последовательности происходит вычисление известных метрик, показывающих её диагностические свойства. Второй режим работы используется в эволюционном алгоритме построения диагностических тестов и вычисляет заданную метрику относительно заданного множества неразличимых неисправностей. Совмещение двух данных режимов в одном алгоритме позволяет эффективно использовать его в качестве ядра при построении автоматизированной системы диагностического моделирования.

В качестве дальнейших исследований можно отметить построение параллельной версии данного метода для многоядерных рабочих станций. Также перспективным является построение автоматизированной диагностической системы, основой которой являются генератор диагностических тестов и программа диагностического моделирования.



2. Грунский И.С. Синтез и идентификация автоматов /Грунский И.С., Козловский В.А. — К.: Наукова думка, 2004. — 245 с.
3. Скобцов Ю.А. Основы эволюционных вычислений /Ю.А. Скобцов. — Донецк: ДонНТУ, 2008. — 326 с.
4. Goldberg D.E. Genetic Algorithm in Search, Optimization, and Machine Learning /D.E. Goldberg. — Boston, MA: Addison-Wesley Longman Publishing Co., 1989. — 412 p.
5. Kirkpatrick S. Optimization by simulating annealing /S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi //Science, 1983. — V.220. — P.671-680.
6. Corno F. GARDA: a Diagnostic ATPG for Large Synchronous Sequential Circuits /F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda //Proc. of IEEE European Design and Test Conference, Paris, March 1999, pp.267-271.
7. Иванов Д.Е. Генетический алгоритм построения диагностических последовательностей цифровых устройств /Д.Е. Иванов //Вісник Східноукраїнського національного університету ім.В. Даля. — 2010. — №10 (152). — С.72-79.
8. Борисевич А.В. Генетический алгоритм для построения диагностического теста сложных цифровых схем /А.В. Борисевич, Р.О. Берзин //Вестник СевГТУ: «Информатика, электроника, связь». — 2006. — Вып. 74. — С.91-96.
9. Zhang Yu. A Diagnostic Test Generation System and a Coverage Metric /Yu Zhang, V.D. Agrawal //Proc. of Int. Test Conf., May 2010. — Pp. 1-9.
10. Иванов Д.Е. Автоматизированная система моделирования и идентификации цифровых устройств АСМИД- Evolution /Д.Е. Иванов //Проблемы информационных технологий. — 2011. — №1 (009). — С.114-124.
11. Скобцов Ю.А. Логическое моделирование и тестирование цифровых устройств /Ю.А. Скобцов, В.Ю. Скобцов. — Донецк: ИПММ НАНУ, ДонНТУ, 2005. — 436 с.
12. Chess B. Creating Small Fault Dictionaries /B. Chess, T. Larrabee //IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Mar. 1980. — Pp.346-356.
13. Lavo D. Making Cause-Effect Cost Effective: Low-resolution Fault Dictionaries /D. Lavo, T. Larrabee //in Proc. International Test Conf., 2001. — Pp.278-286.
14. Pomeranz I. On the Generation of Small Dictionaries for Fault Location /I. Pomeranz, S.M. Reddy //Proc. Intl. Conf. Computer-Aided Design, 1992. — Pp.272-278.
15. Мионов С. В. Генетические алгоритмы для сокращения диагностической информации /С.В. Мионов, Д.В. Сперанский //Автоматика и телемеханика. — 2008. — №7. — С.146-156.
16. Arslan B. Fault dictionary size reduction through test response superposition /B. Arslan, A. Orailoglu //Proceedings of the 2002 IEEE International Conference on Computer Design: VLSI in Computers (ICCD'02), 2002. — Pp.480-485.
17. Rudnick, E.M. Diagnostic Fault Simulation of Sequential Circuits /E.M. Rudnick, W.L. Fuchs, J.H. Patel //Proc. of International Test Conference, 20-24 Sep. 1992. — Pp.178-186.
18. Bodoh D. Diagnostic Fault Simulation for the Failure Analyst /D. Bodoh, A. Blakely, T. Garyet //Conference Proceedings from the 30th International Symposium for Testing and Failure Analysis (ASM International), Oct. 2004. — Pp.181-190.
19. Camurati P. Diagnosis Oriented Test Pattern Generation /P. Camurati, A. Lioy, P. Prinetto, M. Sonza Reorda //Proc. European Design Automation Conf., March 1990. — Pp.470-474.
20. Jou J.-M. A fast and memory-efficient diagnostic fault simulation for sequential circuits /J.-M. Jou, S.-C. Chen //Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design. — IEEE Computer Society Press Los Alamitos, CA, USA, 1994. — Pp.723-726.
21. Chen S.-C. Serial diagnostic fault simulation for synchronous sequential circuits /S.-C. Chen, J.-M. Jou //Integration, the VLSI journal. — Vol. 23. — 1997. — Pp.157-170.
22. Seshadri D. Accelerating diagnostic fault simulation using z-diagnosis and concurrent equivalence identification //B. Seshadri, S. Venkataraman //Proceedings of 24th IEEE VLSI Test Symposium, April 30, 2006. — Pp.385-390.
23. Michael L. High-Performance Diagnostic Fault Simulation on GPUs /M.L. Michael, S. Hsiao //Proceedings of the 2011 Sixteenth IEEE European Test Symposium. — IEEE Computer Society Washington, DC, USA. — 2011. — Pp.210.
24. Kubiak K. Exact Evaluation of Diagnostic Test Resolution /K. Kubiak, S. Parkes, W.K. Fuchs, R. Saleh //Proc. 29th Design Automation Conf., June 1992. — Pp.347-352.
25. Venkataraman S. Rapid Diagnostic Fault simulation of Stuck-at Faults in Sequential Circuits using Compact List /S. Venkataraman, I. Hartanto, W.K. Fuchs, E.M. Rudnick, S. Chakravarty, J.H. Patel //Proc. Design Automation Conf., June 1995. — Pp.133-138.
26. Иванов Д.Е. Параллельное моделирование неисправностей для последовательных схем /Д.Е. Иванов, Ю.А. Скобцов //Искусственный интеллект. — 1999. — №1. — С.44-50.
27. Иванов Д.Е. Параллельный алгоритм моделирования цифровых схем с неисправностями для многоядерных систем с общей памятью /Д.Е. Иванов //Электронное моделирование. — 2011. — Т.33, №1. — С.93-106.