

# ФОРМАЛЬНАЯ ОЦЕНКА ИДЕНТИФИЦИРУЮЩИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ЦИФРОВЫХ УСТРОЙСТВ

УДК 004.3:681.518

**ИВАНОВ Дмитрий Евгеньевич**

д.т.н., доцент старший научный сотрудник отдела теории управляющих систем Института прикладной математики и механики НАН Украины, Донецк; профессор кафедры АСУ, Донецкий национальный технический университет, Донецк

**Научные интересы:** генетические алгоритмы, техническая диагностика, генерация тестов, эволюционные оптимизационные стратегии, моделирование цифровых схем, параллельные вычисления

## ВВЕДЕНИЕ

Проектирование современных высоконадёжных цифровых устройств (ЦУ) – сложный процесс, состоящий из целого ряда этапов. Каждый этап ставит перед собой различные цели: от верификации эквивалентности поведения реализации ЦУ его спецификации до проверки целостности процессов оптимизации логики ЦУ. Поэтому на различных этапах процесса проектирования цифровых устройств применяются различные типы проверяющих входных последовательностей. Это могут быть тестовые последовательности, характеристические, диагностические, верифицирующие эквивалентность и т.д [1]. Будем обобщённо такие последовательности называть идентифицирующими (ИдП).

Одним из интенсивно развивающихся подходов построения ИдП различных классов являются эволюционные алгоритмы [2-3]. Первоначальный импульс исследователям был дан итальянской группой университета г. Турино [4]. Далее аналогичные исследования стали проводиться по всему миру [5-6].

Разработка методов построения ИдП с применением эволюционных вычислений идёт в двух направлениях. Во первых, рассматриваются задачи построения различных классов последовательностей, например, тестовых [4], инициализирующих [7], верификации эквивалентности поведения [8], диагностических (локализующих неисправность) [9]. Во вторых, расширяется набор применяемых эволюционных эвристик. В

частности, используются генетические алгоритмы (ГА) [4, 7-8], алгоритмы симуляции отжига [10-11], муравьиные алгоритмы [6] и т.д. Работа отечественных авторов в этом направлении обобщена в монографии [12].

Основной особенностью всех разрабатываемых эволюционных методов является оценка потенциальных решений-последовательностей на основании моделирования. Именно вид оценочной функции заданной последовательности, способ её построения и скорость вычисления во многом определяют эффективность методов данного рода. Во всех упомянутых работах авторы вводят эвристические функции оценки качества особей, которые следуют из смысловой нагрузки конкретной задачи. Однако ни в одной работе не приводится ни формального описания введённых оценочных функций последовательностей, ни математического аппарата такого описания.

Как было отмечено выше, авторы также проводили исследования в данном направлении. Был разработан целый ряд методов генерации диагностирующих последовательностей цифровых устройств структурного уровня, для которых, в частности, были предложены оценочные функции последовательностей. На основании такого эвристического опыта в данной работе строится формальный математический аппарат оценки качества идентифицирующих последовательностей различных классов. Основная идея формализации заключается в том, что несмотря на различие смысловой нагрузки последовательностей, типы верификаций

требуемых параметров при их построении ограничены и сводятся небольшому числу.

Предложенный математический аппарат был применён при разработке методологии синтеза эволюционных методов диагностирования ЦУ [13] и показал высокую эффективность. Однако он, безусловно, будет полезен и в других методах построения диагностических последовательностей.

### ТИПЫ ИДЕНТИФИЦИРУЮЩИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ЦИФРОВЫХ УСТРОЙСТВ

Дадим формальные определения объекта исследования. В зависимости от задачи и удобства подхода её решения ЦУ может рассматриваться как правильная логическая сеть элементов структурного уровня, либо как абстрактный автомат.

Пусть рассматривается работа синхронного последовательностного ЦУ  $A_0$  в трёхзначном алфавите  $E_3 = \{0, 1, u\}$ . В заданном ЦУ  $A_0$ :  $N_{вх}$  - число внешних входов,  $N_{вых}$  - число внешних выходов,  $N_{эл}$  - число логических элементов,  $N_{мп}$  - число элементов состояний (псевдовыходов/псевдовходов). Пусть  $Q$  - множество всевозможных состояний последовательностного ЦУ, включая неопределённые. Тогда  $Q = \{0, 1, u\}^{N_{мп}}$ . Пусть  $Z$  - множество всех определённых состояний ЦУ:  $Z = \{0, 1\}^{N_{мп}}$ . Легко видеть, что  $Z \subset Q$ . Множество всех возможных входных последовательностей  $S_i$  образуют множество  $\Sigma$ , причём элементы  $S_i$  кодируются элементами из  $E_3$ . Также  $Z_u = (uu...u)$  - начальное полностью неопределённое состояние ЦУ,  $Z_u \in Q$ . Отображение  $\delta: Q \times \Sigma \rightarrow Q$  показывает все достижимые состояния ЦУ, когда на его вход поступают  $S_i \in \Sigma$  при работе в алфавите  $E_3$ . В автоматной терминологии:  $\delta$  - функция перехода при работе в алфавите  $E_3$ .

Дадим формальные определения основных типов ИДП ЦУ структурного уровня.

**Определение 1.** Для ЦУ  $A_0$  с функцией перехода  $\delta$  последовательность  $S \in \Sigma$  называется последовательностью достижения состояния (ПДС, установочной)  $Z_{кон}$ , если для  $\forall \alpha \in Q$  будет выполняться:  $Z_{кон} = F(\alpha, S)$ .

Возможна постановка задачи, когда задано как конечное  $Z_{кон}$ , так и начальное  $Z_{нач}$  состояние ЦУ, либо множество начальных состояний  $Q_{старт}$ .

**Определение 2.** Последовательность  $S \in \Sigma$  называется логически инициализирующей (ИИП) для заданного ЦУ  $A_0$  с функцией перехода  $\delta$ , если финальное состояние  $Z_i = F(Z_u, S) \in Z$ , т.е. в нём определены все элементы состояний.

Различие между ИИП и ПДС заключается в следующем. В задаче построения ИИП речь идёт именно о стартовом режиме работы ЦУ, тогда как установочные являются вспомогательными в различных алгоритмах диагностирования ЦУ.

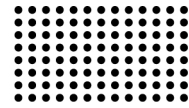
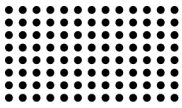
В [1, с.390] вводятся понятия  $R$ - и  $S$ -последовательностей, которые также помогают решать задачу перевода ЦУ в требуемое состояние.

**Определение 3** Последовательность называется сбрасывающей  $R_i$  (устанавливающей  $S_i$ ), если после её приложения ко входам ЦУ  $A_0$ , находящемуся в полностью неопределённом состоянии  $Z_u = (uu...u)$ ,  $i$ -й элемент состояния  $z_i \in Z$  получит значение 0 (1) в алфавите моделирования  $E_3$ .

Данные  $R$ - и  $S$ -последовательности называются последовательностями типа  $A$ . Если же потребовать, что сброс (установка) значения на  $i$ -м элементе состояния достигается для ЦУ, в котором некоторые начальные элементы состояний уже фиксированы (имеют значение 0 или 1) и не изменяются, то такие последовательности называются  $R$ - и  $S$ -последовательностями типа  $B$ .

Часто применяются также последовательности, проверяющие эквивалентность поведения заданных ЦУ, что связано, например, с проверкой корректности процесса их оптимизации. Определения эквивалентности наиболее часто даются для трёхзначных логик, поскольку именно в трёхзначном алфавите чаще всего происходит моделирование.

**Определение 4.** ЦУ  $A_1$  называется трёхзначной не-изменяющей заменой (ТНЗ) для ЦУ  $A_2$  тогда и только тогда, когда для произвольной последовательности  $S$  выходные реакции  $A_1(Z_u, S)$  покрывают выходные реакции  $A_2(Z_u, S)$ .



При этом значение сигнала  $v_1$  называется покрывающим для значения сигнала  $v_2$ , если пара  $(v_1, v_2)$  принимает только следующие значения:  $(0, 0)$ ,  $(1, 1)$ ,  $(u, u)$ ,  $(u, 0)$  и  $(u, 1)$ , где  $1, 0, u \in E_3$ .

Видно, что определение эквивалентности специально сужается для таких ЦУ, которые начинают работу из полностью неопределённого состояния  $Z_u$ , что является очень распространённым случаем на практике.

**Определение 5.** ЦУ  $A_1$  является трёхзначным эквивалентом (ТЭ) для ЦУ  $A_2$  тогда и только тогда, когда при их при старте из неопределённого состояния  $Z_u$  для любой входной последовательности  $S$  пары выходных реакций на неё  $(A_1(Z_u, S), A_2(Z_u, S))$  принадлежат множеству  $\{(0, 0), (1, 1), (u, u)\}$ .

Отметим, что такое построение определений ТНЗ и ТЭ позволяет переформулировать задачу верификации эквивалентности: вместо формального доказательства эквивалентности двух заданных ЦУ можно строить последовательности, которые показывают их неэквивалентность в заданном смысле.

**Определение 6.** Входная последовательность  $S$ , выходные реакции на которую заданных устройств  $A_0$  и  $A_i$  различны, называется проверяющей, т.е.  $A_0(S) \neq A_i(S)$ .

Данное определение служит основным при решении задачи построения проверяющих тестов, которая может быть сведена к итеративной задаче построения проверяющей последовательности для исправного  $A_0$  и неисправного  $A_i$ , моделирующего поведение ЦУ в присутствии некоторой неисправности из заданного класса  $f_i \in F = \{f_1, \dots, f_n\}$

**Определение 7.** Пусть задано исправное ЦУ  $A_0$  и класс неисправностей  $F = \{f_1, \dots, f_n\}$ , который порождает класс неисправных устройств  $A = \{A_1, \dots, A_n\}$ . Входная последовательность  $S$ , которая может отличить поведение произвольного устройства  $A_i$  от поведения исправного  $A_0$ , а также поведения всех остальных неисправных устройств, называется диагностической.

Фактически, диагностическая последовательность позволяет разбить весь класс неисправностей на классы эквивалентных неисправностей, при этом мощность каждого такого класса равна единице [9, 12]. На прак-

тике задача построения диагностических последовательностей решается итеративным построением различающих последовательностей.

**Определение 8.** Для заданного устройства  $A_0$  и заданного класса неисправностей  $F = \{f_1, \dots, f_n\}$  последовательность  $S$  называется различающей, если существует хотя бы одна неисправность  $f_j \in F$  такая, что  $A_j(S) \neq A_0(S)$  (для  $i = \overline{1, n}$  кроме  $i = j$ ) хотя бы для одного входного вектора:

$$\exists f_j \in F : A_j(S) \neq A_0(S), i = \overline{1, n}, i \neq j \quad (1)$$

Последнее определение показывает, что при существовании последовательности  $S$  с указанными свойствами класс неисправностей  $F$  разбивается на два класса неразличимых неисправностей. В первый класс входят все неисправности  $A_j$ , для которых выполнено условие в определении; во второй класс - все остальные. Часто такую последовательность  $S$  называют различающей, поскольку она делит один класс неотличимых неисправностей  $F$  на несколько. Неформально такую последовательность также можно назвать диагностической.

### МАТЕМАТИЧЕСКИЙ АППАРАТ ОЦЕНКИ ИДЕНТИФИЦИРУЮЩИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Определения, данные в предыдущем разделе, позволяют сделать вывод, что для оценки качества ИдП в основном отслеживаются поведение значений сигналов по множествам выходных линий и линий состояний. При практической реализации [12] применяются дополнительные эвристики. В частности, используется информация структурного уровня о поведении логических вентилях или поведении в контрольных точках. Именно эта информация даёт преимущество структурных методов над абстрактными. Также одними из основных являются активность вентилях в ЦУ и различающая активность для нескольких ЦУ. Обычно, чем выше значение данных параметров – тем лучше качество оцениваемой последовательности. Также часто в оценочную функцию включают длину последовательности.

Для математической формализации качества заданной последовательности введём следующие функ-

ции поведения ЦУ структурного уровня. Пусть функция  $g(X_j, A_0)$  показывает значение на выходе компоненты ЦУ  $g$  при завершении моделирования ЦУ  $A_0$  на входном наборе  $X_j = S[j]$  последовательности  $S$ .

**Определение 9.** Функция установки в ноль  $U^{(0)}(g, X_j, A_0)$  равна 1, если после моделирования устройства  $A_0$  на входном наборе  $X_j$  выход элемента  $g$  равен нулю:

$$U^{(0)}(g, X_j, A_0) = 1 \Leftrightarrow g(X_j, A_0) = 0 \quad (2)$$

**Определение 10.** Функция установки в единицу  $U^{(1)}(g, X_j, A_0)$  равна 1, если после моделирования устройства  $A_0$  на входном наборе  $X_j$  выход элемента  $g$  равен единице:

$$U^{(1)}(g, X_j, A_0) = 1 \Leftrightarrow g(X_j, A_0) = 1 \quad (3)$$

Тогда установка фиксированного значения 0 или 1 вместо значения  $u \in E_3$  на выходе некоторого элемента  $g$  определяется выражением:

$$U^1(g, X_j, A_0) \vee U^0(g, X_j, A_0) \quad (4)$$

Ценность того факта, что данный элемент  $g$  получил определённое значение на выходе задаётся параметром управляемости  $H_g$ . Он должен быть учтён в виде веса для (4):

$$H_g \cdot (U^1(g, X_j, A_0) \vee U^0(g, X_j, A_0)) \quad (5)$$

При построении оценочной функции можно использовать как (4) так и (5) в том случае, если известны значения  $H_g$ .

**Определение 11.** Функция активности вектора  $X_j$  (число событий) показывает число изменения значений сигналов в сравнении с предыдущим входным набором по выбранному множеству  $G$ :

$$N_{cob}(A_0, X_j) = \sum_{g \in G} ((U^{(1)}(g, X_{j-1}, A_0) = U^{(0)}(g, X_j, A_0)) \vee (U^{(0)}(g, X_{j-1}, A_0) = U^{(1)}(g, X_j, A_0))) \quad (6)$$

где  $G$  - множество рассматриваемых элементов ЦУ.

**Определение 12.** Функция активности входной последовательности  $S$  есть суммарная активность для каждого входного набора данной последовательности:

$$N_{cob}(A_0, S) = \sum_{i=1}^{длина(S)} N_{cob}(A_0, X_j) \quad (7)$$

**Определение 13.** Функция различия показывает эквивалентность/неэквивалентность поведения соответствующих компонент устройств  $A_1$  и  $A_2$

$$r_2(g, X_j, A_1, A_2) = g(X_j, A_1) \oplus g(X_j, A_2) \quad (8)$$

где: 2 – индекс в  $r()$ , показывающий значность алфавита моделирования;  $g(X_j, A_k)$  - выходное значение на выходе комбинационного блока  $g$  при завершении моделирования ЦУ  $A_k, k = \overline{1, 2}$  на входном наборе  $X_j$ .

Функция различия  $r_2()$  для двухзначного алфавита моделирования также выражается через введённые ранее функции достижения  $U^0$  и  $U^1$ :

$$r_2(g, X_j, A_1, A_2) = (U^{(0)}(g, X_j, A_1) = U^{(1)}(g, X_j, A_2)) \vee (U^{(1)}(g, X_j, A_1) = U^{(0)}(g, X_j, A_2)) \quad (9)$$

Активность на выходе различных компонент ЦУ по разному может влиять на активность его выходов, что задаётся наблюдаемостью  $I_g$ :

$$r_2(g, X_j, A_1, A_2) = I_g \cdot (U^{(0)}(g, X_j, A_1) = U^{(1)}(g, X_j, A_2)) \vee (U^{(1)}(g, X_j, A_1) = U^{(0)}(g, X_j, A_2)) \quad (10)$$

Замечания о применении параметра  $I_g$  такие же, как и для  $H_g$ .

Вычисление значений введённых функции поведения выполняется после каждого такта модельного времени путём сравнения массивов значений сигналов на линиях ЦУ  $A_1$  и  $A_2$ .

**ПРИМЕРЫ ПОСТРОЕНИЯ ОЦЕНОК  
В ПРАКТИЧЕСКИХ ЗАДАЧАХ**

В данном разделе на основании введенных функций поведения ЦУ будет показано построение оценочных функций ИдП для ряда практических задач.

Начнем с задачи достижения состояния ЦУ. Семантически оценочная функция показывает, насколько близко текущее состояние  $Z_{тек}$  моделируемого ЦУ  $A_0$ , получаемое после подачи последовательности  $S$ , находится от требуемого состояния  $Z_{кон}$ . Поскольку считается, что граф переходов ЦУ  $A_0$  не известен, то для формализации оценки будем использовать расстояние по Хэммингу для двоичного представления  $Z_{тек} = \delta(Z_{нач}, S)$  и  $Z_{кон}$ , показывающее число совпадений элементов состояний. Таким образом, для заданного ЦУ  $A_0$  и заданной входной последовательности  $S$  оценка вычисляется на основании результатов исправного моделирования:

$$O(A_0, S, Z_{нач}, Z_{кон}) = O(Z_{тек}(A_0, S, Z_{нач}), Z_{кон}) = \sum_{Z_{тек}[i]=Z_{кон}[i]} 1, \quad (11)$$

где  $Z[i]$  показывает значение  $i$ -го элемента состояния,  $i = 1, \dots, m$ .

Выражение (11) также можно представить через функции установки значений, введенные ранее. Поскольку  $Z_{тек}$  определяется как достижение определенного значения 0 или 1 для элементов  $g \in Z$ , то значения компонент  $g_{кон}$  определяют состояние  $Z_{кон}$ , а последнее состояние ЦУ задаётся при  $j = \text{длина}(S) - 1$ , то с учётом (2)-(3) можно записать:

$$O(A_0, S, Z_{нач}, Z_{кон}) = \sum_{g \in Z} ((U^{(0)}(g, X_j, A_0, Z_{нач}) = g_{кон}) \vee \vee (U^1(g, X_j, A_0, Z_{нач}) = g_{кон})) \quad (12)$$

где:  $Z$  - множество элементов состояний ЦУ.

Задача построения  $R$ - и  $S$ -последовательностей в соответствии с определением 3 является частным случаем задачи построения ПДС, поэтому в качестве оценки может выступать функция (12) с соответствующим упрощением.

При решении задачи инициализации ЦУ оценка в виде (11) обычно расширяется параметрами активности ЦУ и длины последовательности. С их учётом она принимает вид:

$$O(A_0, S) = O(n_1(A_0, S), n_2(A_0, S), n_3(S)) = (c_1 * n_1(A_0, S) + c_2 * n_2(A_0, S)) * c_3^{n_3(S)}, \quad (13)$$

где:  $c_1, c_2, c_3$  - нормализующие константы;  $n_1$  - отношение числа элементов состояний с определённым значением сигнала из  $E_2 = \{0, 1\}$  к общему числу элементов состояний  $N_{mp}$ ;  $n_2$  - число событий при моделировании поведения или активность ЦУ;  $n_3$  - длина особи  $S$ .

Видно, что оценочная функция последовательности  $S$  является многокритериальной. Для параметров  $n_1$  и  $n_2$  она строится по аддитивному принципу [14]. Параметр  $n_3$  входит в функцию  $O()$  в виде показателя степени. Важность каждого из критериев в общей оценке определяется нормирующими константами  $c_1$  и  $c_2$ , которые показывают вес параметров инициализации триггеров и активности элементов в устройстве. Очевидно, что первый из них  $n_1(A_0, S)$  является более ценным в смысле решения задачи, тогда как на параметр  $n_2(A_0, S)$  следует ориентироваться в том случае, когда инициализация триггеров приостановилась. Коэффициент  $c_3^{n_3(S)}$  необходим для того, чтобы более высокую оценку получали более короткие последовательности  $S$ . Это будет достигаться при выборе  $0 < c_3 < 1$ . Следует отметить, что при  $c_3 \ll 1$  глубина поиска увеличится, существенно замедляя поиск. Поэтому константу  $c_3$  следует выбирать очень близкой к 1.

Параметры  $n_1$  и  $n_2$  также можно выразить через введенные функции. Для  $n_1$  можно записать:

$$n_1 = \left( \sum_{g \in Z} (U^{(1)}(g, X_j, A_0) \vee U^{(0)}(g, X_j, A_0)) \right) / N_{mp}, \quad (14)$$

где  $Z$  - множество линий элементов состояний ЦУ.

Параметр  $n_2$  в точности совпадает с функцией активности (7).

Отметим, что функция оценки (11) в задаче достижения заданного состояния ЦУ по аналогии с (13)

может быть расширена включением параметров активности ЦУ и длины последовательности.

При верификации эквивалентности поведения двух ЦУ семантическая нагрузка свойства строящейся

$$O(A_1, A_2, S) = \sum_{j=1}^{\text{длина}(S)} ((k_1 \cdot n_1(A_1, A_2, X_j) + k_2 \cdot n_2(A_1, A_2, X_j) + k_3 \cdot n_3(A_1, A_2, X_j)), \quad (15)$$

где:  $k_1 - k_3$  - нормализующие константы;  $n_1 - n_3$  - числовые параметры, показывающие суммарное различие поведения  $A_1$  и  $A_2$  по множествам внешних выходов, линий состояний и комбинационных блоков соответственно.

Эвристические параметры  $n_1 - n_3$  строятся на основании функции различия поведения компонент:

$$n_i(A_1, A_2, X_j) = \sum_{g \in Y} r(g, X_j, A_1, A_2) \quad (16)$$

где в качестве множества суммирования выступают множества внешних выходов, линий состояний и комбинационных блоков соответственно. Функцию в виде (16) будем называть различающей активностью двух заданных ЦУ по выбранному множеству компонент.

Определение 13 функции  $r_2()$  дано в алфавите  $E_2$  и требует расширения на трёхзначную логику с учётом определений 4 и 5. Для алгоритма верификации ТНЗ функция  $r()$  будет принимать значений 1, если пара выходов элементов  $g_1(X_j, A_1)$  и  $g_2(X_j, A_2)$  принадлежит запрещённому множеству  $D_1 = \{(0,1), (1,0)\}$ :

$$r'_3(g, x_j, A_1, A_2) = \begin{cases} 1, & \text{если } (g(x_j, A_1), g(x_j, A_2)) \in D_1; \\ 0, & \text{в противном случае.} \end{cases} \quad (17)$$

Для алгоритма верификации трёхзначной эквивалентности функция  $r()$  будет принимать значений 1, если пара выходов элементов  $g_1(X_j, A_1)$  и  $g_2(X_j, A_2)$  принадлежит запрещённому множеству  $D_2 = \{(0,1), (1,0), (u,0), (u,1), (0,u), (1,u)\}$ :

последовательности заключается в том, чтобы породить наибольшее различие в поведении верифицируемых ЦУ  $A_1$  и  $A_2$ :

$$r''_3(g, X_j, A_1, A_2) = \begin{cases} 1, & \text{если } (g(X_j, A_1), g(X_j, A_2)) \in D_2; \\ 0, & \text{в противном случае.} \end{cases} \quad (18)$$

Таким образом, применяя функции различия в виде (17) либо (18) в функции оценки (16) мы будем строить различающие последовательности для определений ТНЗ и ТЭ соответственно.

Задача построения тестов [12] представляется как итеративная задача построения проверяющей последовательности для текущей целевой неисправности  $f_{цел}$ . Тогда оценка некоторой входной последовательности  $S$  тем выше, чем большую различающую активность для устройств  $A_0$  и  $A_f$  (устройство в присутствии неисправности  $f_{цел}$ ) она произведёт в процессе моделирования:

$$O(A_0, f_{цел}, S) = \sum_{i=1}^{\text{длина}(S)} \sum_{g \in G} r(g, X_i, A_0, A_f) \quad (19)$$

где  $X_i$  -  $i$ -й входной набор последовательности  $S$ ,  $G$  - множество логических элементов (контрольных точек) ЦУ.

Видно, что оценка в виде (19) является частным случаем оценки (15), в которой убраны компоненты различающей активности множеств внешних выходов и элементов состояний. Различие по внешним выходам не учитывается, поскольку по результатам моделирования сразу видно, является ли последовательность проверяющей. Компонент различия по множеству элементов состояний отсутствует, поскольку оценка в виде (19) производится для последовательностей, которые заведомо переводят устройства  $A_0$  и  $A_f$  в различные состояния.

Видно также, что (19) полностью соответствует различающей активности (16), вычисленной для всей последовательности  $S$ .

В качестве функционала  $O(A_0, f_{цел}, S)$  в (19) часто используется не суммирование, а выбор максимального значения [4]:

$$O(A_0, f_{цел}, S) = \max_i \sum_{g \in G} r(g, X_i, A_0, A_f) \quad (20)$$

Аналогичным образом могут быть переписаны функции оценки (11), (12) и (15). Конкретный вид оценки, применяемый в реализации метода, устанавливается на основании машинных экспериментов.

Рассмотрим построение функции оценки в методе построения диагностических тестов [9, 12]. Он заключается в итеративном построении различающих последовательностей для множеств неразличимых неисправ-

ностей. Для последовательности  $S$  и класса неисправностей  $F_j$  легко определить, является ли  $S$  диагностической (определения 7-8). Для этого достаточно выполнить диагностическое моделирование [15]. Однако оценить, насколько близко данная последовательность  $S$  подошла к решению такой задачи, гораздо труднее. В задачах верификации эквивалентности и построения тестов оценочная функция последовательности  $S$  строилась на основании различающей активности (16) устройств: чем больше различных значений на контрольных множествах в двух сравниваемых ЦУ, тем выше оценка входной последовательности. Выберем аналогичный подход и при решении данной задачи. Общая оценка последовательности  $S$  строится как сумма оценок каждого входящего в неё вектора  $X_i$ :

$$O(A_0, F_j, S) = \sum_{i=1}^{длина(S)} O(A_0, F_j, X_i) = \sum_{i=1}^{длина(S)} (c_1 \cdot N_1(X_i, A_0, F_j) + c_2 \cdot N_2(X_i, A_0, F_j) + c_3 \cdot N_3(X_i, A_0, F_j)), \quad (21)$$

где:  $X_i$  -  $i$ -й набор последовательности  $S$ ;  $c_1$  -  $c_3$  - нормализующие константы;  $N_1$  -  $N_3$  - параметры различающей активности.

В отличие от задач верификации эквивалентности и построения тестов здесь происходит сравнение поведения не двух ЦУ, а фактически целого множества ЦУ, задаваемого множеством неисправностей  $F_j$ . Поэтому параметры  $N_1$ ,  $N_2$  и  $N_3$ , несут несколько другую смысловую нагрузку.

$N_1$  - число внешних выходов с различными значениями в неисправных ЦУ, соответствующих неисправностям, входящим в класс неразличимости  $F_j$ :

$$N_1(X_i, A_0, F_j) = \sum_{g \in Y} R(g, X_i, A_0, F_j) \quad (22)$$

где  $Y$  - множество внешних выходов ЦУ.

$N_2$  - число элементов состояний с различными значениями в неисправных ЦУ, соответствующих неисправностям, входящим в класс неразличимости  $F_j$ :

$$N_2(X_i, A_0, F_j) = \sum_{g \in Z} R(g, X_i, A_0, F_j) \quad (23)$$

где  $Z$  - множество линий состояний ЦУ.

$N_3$  - число выходов вентиляей (комбинационных блоков, контрольных точек) с различными значениями в неисправных ЦУ, которые соответствуют неисправностям, входящим в текущий класс неразличимости  $F_j$ :

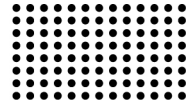
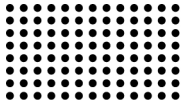
$$N_3(X_i, A_0, F_j) = \sum_{g \in G} R(g, X_i, A_0, F_j) \quad (24)$$

где  $G$  - множество линий комбинационных блоков ЦУ.

Соответственно, функции различия  $R()$  в (22)-(24) переопределяются для множества сравниваемых ЦУ следующим образом:

$$R(g, X_i, A_0, F_j) = \begin{cases} 0, & \text{если выходы элементов } g \text{ множества ЦУ,} \\ & \text{которые принадлежат классу } F_j, \text{ одинаковы} \\ & \text{после подачи набора } X_i; \\ 1, & \text{иначе.} \end{cases} \quad (25)$$

Функция различия  $r()$  двух устройств (8) является частным случаем функции различия  $R()$  для множеств



ва устройств, когда верифицируется поведение пары ЦУ. Также может быть легко показано, что функции различия  $R()$  выражаются через функции установки значений  $U^{(0)}$  и  $U^{(1)}$ .

Видно также, что функции оценки (15) и (19) в методах верификации эквивалентности и построения тестов являются частными случаями оценки в виде (21).

По аналогии с (20) оценка (21) также может быть выражена через поиск максимального элемента.

### ВЫВОДЫ

В работе предложен формальный математический аппарат оценки качества диагностирующих последовательностей цифровых устройств для логического уровня представления. С этой целью введены понятия функций поведения ЦУ структурного уровня: функция

установки значения компоненты, функции активности и различия по множеству элементов. Данные функции отражают поведение устройства на структурном уровне и определяют ту дополнительную информацию, которая даёт преимущество структурных методов диагностирования над абстрактными. Показано, что все динамические параметры оценочных функций диагностирующих последовательностей выражаются через введённые функции поведения.

Применение построенного математического аппарата даёт возможность при построении оценочных функций формализовать информацию о поведении устройств на структурном уровне, более точно оценивать строящиеся последовательности и, следовательно, улучшить сходимость методов генерации диагностирующих последовательностей различных типов.

### ЛИТЕРАТУРА:

1. Skobtsov Yu.A. Logicheskoe modelirovanie i testirovanie tsifrovyykh ustroystv / Yu.A. Skobtsov, V.Yu. Skobtsov.- Donetsk:IPMM NANU, DonNTU, 2005.- 436s.
2. Goldberg D.E. Genetic Algorithm in Search, Optimization, and Machine Learning.- Boston, MA: Addison-Wesley Longman Publishing Co.- 1989.- 412p.
3. Skobtsov Yu.A. Osnovy evolyutsionnykh vyichisleniy.- Donetsk: DonNTU, 2008.- 326s.
4. Corno F. Experiences in the use of evolutionary techniques for testing digital circuits / F. Corno, M. Sonza Reorda, M. Rebaudengo // Proc. of Conf. Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA.- 1998.- P.128-139.
5. Rudnick E.M. Sequential Circuit Test Generation in a Genetic Algorithm Framework / E.M. Rudnick, J.H. Patel, G.S. Greenstein, T.M. Niermann // Proc. Design Automation Conf.- 1994.- P.698-704.
6. Xiaojing H., Zhengxiang S. Ant Colony Optimizations for Initialization of synchronous sequential circuits // IEEE Circuits and Systems International Conf., 2009. – P. 5–18.
7. Corno F. A Genetic Algorithm for the Computation of Initialization Sequences for Synchronous Sequential Circuits / F. Corno, P. Prinetto, M. Rebaudengo etc. // Proceeding ATS '01 Proceedings of the 10th Anniversary Compendium of Papers from Asian Test Symposium 1992-2001.- 2001.- P.213.
8. Corno F. Evolutionary Simulation-Based Validation / F. Corno, M. Sonza Reorda, G. Squillero // International Journal on Artificial Intelligence Tools (IJAIT).- 2004.- Vol.14, 1-2, Dec.- P. 897-916.
9. 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.
10. Ivanov D.E. Primenenie algoritmov simulyatsii otzhiga v zadachah identifikatsii tsifrovyykh shem // Vestnik Natsionalnogo tekhnicheskogo universiteta «Harkovskiy politekhnicheskiy institut». Sbornik nauchnykh trudov. Tematicheskyy vyipusk: Informatika i modelirovanie.– Harkov: NTU "HPI", 2011.– # 17.– С.60-69.
11. Corno F. SAARA: a simulated annealing algorithm for test pattern generation for digital circuits / F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda // Proceedings of the 1997 ACM symposium on Applied computing, San Jose, California.- 1997.- P.228-232.
12. Ivanov D.E. Geneticheskie algoritmyi postroeniya vhodnykh identifikatsionnykh posledovatel'nostey tsifrovyykh ustroystv / D.E. Ivanov. Donetsk, 2012. 240s.
13. Ivanov D.E. Metodologiya sinteza evolyutsionnykh algoritmov postroeniya identifikatsionnykh posledovatel'nostey tsifrovyykh ustroystv / D.E. Ivanov // «Radloelektronika i komp'yuternye sistemy».- 2013.- #5(64).- S.156-161.
14. Luke S. Essentials of Metaheuristics.- George Mason University Press, USA.- 2009.- 239p.
15. Ivanov D.E. Algoritmy diagnosticheskogo modelirovaniya SBIS / D.E. Ivanov // Problemy informatsionnykh tekhnologiy.- 2012.- #1 (011).- S.107-114.