

Рис. 3. Зависимости светлоты от времени

**Козяев Леонид Леонидович**, аспирант кафедры ПО ЭВМ ХТУРЭ, системный администратор АОЗТ “Экоп-род АТ”. Научные интересы: изучение зрительных свойств человека. Хобби: работа за компьютером, спорт. Адрес: Украина, 342300, Донецкая обл., г. Волноваха, ул. Менделеева, 35, кв. 11, тел. (06244) 4-10-36, 4-16-41.

УДК 681.325:519.613

## ГЕНЕРАЦИЯ ТЕСТОВ ДЛЯ КОНЕЧНЫХ АВТОМАТОВ, ЗАДАННЫХ ГРАФ-СХЕМАМИ АЛГОРИТМОВ

ХАХАНОВ В.И., СЫСЕНКО И.Ю., ЧАМЯН А.Л.

Описывается метод генерации тестов для конечных автоматов, поведение которых задается в форме граф-схем алгоритмов (ГСА). Проектируемые тесты ориентированы на проверку введенного класса дефектов — одиночных неисправностей переходов (ОИП). Задача построения тестов сводится к однократному обходу всех вершин ориентированного графа переходов.

### 1. Введение

Проблема генерации тестов для цифровых конечных автоматов сводится к выбору способа пост-

роения проверяющих последовательностей в зависимости от формы представления модели объекта. Реальность такова, что разработчику тестов могут быть доступны следующие формы описания поведения автомата [1-8]:

1. Функциональная схема с отсутствием макроалгоритма ее работы на уровне ГСА или таблицы переходов. Такая форма задает асинхронную модель устройства с глобальными обратными связями, где примитивами выступают комбинационные или последовательностные элементы.

2. Алгоритмическое описание поведения объекта в виде графа алгоритма, бинарной диаграммы решений (граф-схемы алгоритма), программного модуля на одном из алгоритмических языков или языков описания аппаратуры типа VHDL.

3. Описание поведения объекта на системном уровне, представленного взаимосвязанными компонентами, которые могут быть заданы в виде упомянутых ранее форм или вербальным описанием закона функционирования объекта и его частей.

## 2. Алфавит описания автоматных моделей

Для задания моделей конечных автоматов будем использовать табличное представление граф-схем алгоритмов, как наиболее технологичную форму восприятия информации пользователем и компьютером, которая основывается на кубическом исчислении – математическом аппарате компактного описания цифровых структур.

Основной недостаток табличного представления конечных автоматов – размерность – может быть сведен к минимуму введением избыточности в алфавит кодирования состояний булевых переменных. Алгебра Кантора есть алфавит описания автоматных переменных в одном временном фрейме:

$$A^1 = \{0, 1, X = \{0, 1\}, \emptyset(U)\},$$

где для обозначения высокого импеданса далее будет введен символ  $Z = \emptyset$ .

С позиции теории множеств  $A^1$  – это замкнутый относительно операций пересечения, объединения, дополнения булеан, образованный на универсуме примитивов  $\{0, 1\}$  определением множества всех его подмножеств. Упомянутые операции – отображение логических функций: конъюнкции, дизъюнкции, отрицания в теоретико-множественном алфавите.

Основным компонентом двухтактного кубического исчисления (ДКИ) является алфавит  $A^X$ , который задает все возможные переходы двухтактной автоматной переменной в моменты  $(t-1, t)$  универсумом примитивов  $Y = \{00, 01, 10, 11\}$ . Определение замкнутого алфавита связано со взятием булеана уже на универсуме  $Y$ . Мощность порождаемого таким образом алфавита определяется выражением  $K = 2^n$ , где  $n$  – количество примитивов в семействе  $Y$ . В частности, 16 символов двухтактного алфавита имеют следующий вид [1]:

$A^X = \{Q=00, E=01, H=10, J=11, O=\{Q, H\}, I=\{E, J\}, A=\{Q, E\}, B=\{H, J\}, S=\{Q, J\}, P=\{E, H\}, C=\{E, H, J\}, F=\{Q, H, J\}, L=\{Q, E, J\}, V=\{Q, E, H\}, Y=\{Q, E, H, J\}, U\}$ .

Символ  $U$  обозначает пустое множество  $\emptyset$  и служит для замыкания алфавита относительно теоретико-множественных операций, для его связи с математическими структурами более высокой иерархии и построения систем доказательств. Для подтверждения замкнутости  $A^X$  здесь не приводятся операции пересечения, объединения, дополнения, поскольку данное свойство очевидно по построению.

Имея алфавиты  $A^1, A^X$ , определяющие состояния переменных в моменты  $\langle t \rangle, \langle t-1, t \rangle$ , и в соответствии с принципом симметрии вводим множество  $A^0 = \{G, T, K = \{G, T\}, N\}$ , аналогичное символам  $A^1$ , но служащее для описания автоматных переменных в момент  $\langle t-1 \rangle$ . Замкнутость алфавита гарантируется выполнением бинарных операций пересечения и объединения, унарной операции дополнения. Структурная схема образования всех символов алфавита на двух временных фреймах представлена на рис. 1.

Из трех независимых алфавитов, имеющих различные свойства, создается один универсальный. Объединяющим форматом для них служат два временных фрейма  $\langle t-1, t \rangle$ . Приведение однократных символов  $a_i^0 \in A^0, a_i^1 \in A^1$  к двухтактной форме выполняется с помощью символа  $Z$ , который эквивалентен пустому множеству  $U$  в одном автоматном такте, но "чуть-чуть" полнее последнего и задает описание переменной в состоянии высокого импе-

данса или высокого выходного сопротивления. С помощью буквы  $Z$  форматы однократных символов принимают вид:  $a_i^0 = \langle a, Z \rangle, a_i^1 = \langle Z, a \rangle$ , где  $a$  – однократные булевы состояния из множества  $A = \{0, 1, X = \{0, 1\}, Z\}$ , абстрагированные от времени. При этом символы пустых множеств  $U, Z, N$  на двух автоматных тактах определены в виде

$$Z = UZ; N = ZU; U = UU; UU = ZZ.$$

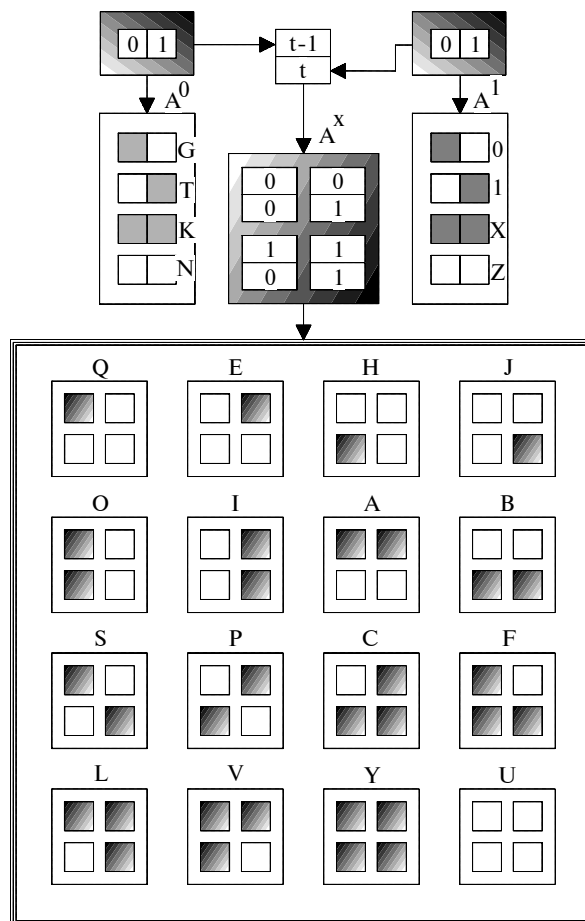


Рис. 1. Диаграммы образования алфавита

Состояние  $ZZ$  будем считать неустойчивым пустым множеством, которое на двух автоматных тактах переходит в стабильную "пустоту", определяемую символом  $U$ . С учетом приведенных равенств отношения введенных символов пустых множеств регулируются тождествами:

$$Z \cap N = U, \quad Z \cup N = ZZ = U,$$

$$Z \cap U = U, \quad Z \cup U = Z,$$

$$N \cap U = U, \quad N \cup U = N.$$

По определению на каждом множестве  $A^0, A^1, A^X$  выполняются все аксиомы и тождества абстрактной математической решетки с дополнениями, такие как ассоциативность, коммутативность, дистрибутивность, идемпотентность, действия с универсумом и пустым множеством, двойного дополнения, правила де-Моргана, поглощения.

## 3. Тестирование неисправностей управляющего автомата

В современных цифровых изделиях компьютерных систем принципиальные схемы модулей и плат являются вполне доступными, но функциональные блок-схемы и алгоритмы, как правило, недоступны

широкому кругу пользователей и специалистов. Это связано с невозможностью практически выполнить анализ схемы в целях понимания функционального назначения. Кроме того, очень просто уяснить или проверить работу упомянутой схемы, если существует алгоритм ее функционирования. Аналогичная ситуация сложилась и в технической диагностике, но по другой причине. Разработчик, проектируя изделие, не заботится о диагностическом обеспечении, а его конечным продуктом является твердая копия принципиальной схемы и алгоритм ее функционирования в голове. Далее схема передается диагносту, который пытается разгадать "кроссворд" проектировщика — определить алгоритм работы устройства — и построить тест. Самое удивительное, что ему это часто удается, вопреки логике. Как должно быть? Идеальное решение задачи — совмещение функций проектировщика и диагноста в одном лице. Следствием такого объединения является знание цифровой схемы и ее закона функционирования, представленной в форме ГСА, таблицы переходов и поведенческой модели, которые служат основой для излагаемого далее метода генерации тестов.

Предлагается модель одиночных неисправностей переходов конечного автомата, построение теста и оценка его полноты. Традиционно модель автономного объекта представляется управляющим и операционным автоматами, как показано на рис. 2.



Рис. 2. Модель операционного устройства (ОС — оповестительные сигналы; ОП — операнды; РО — результат операции; КОП — код операции; ИС — индикация состояния; ИЦИ — инициирующие сигналы)

В физической реализации наиболее сложным для тестирования является блок управляющего автомата (УА), содержащий элементы памяти и комбинационные схемы в виде функций возбуждения. Построение теста для такого блока представляет собой проблему. Определим для УА логическую модель дефектов.

**Определение.** Одиночной неисправностью перехода  $P_{ij}/P_{ir}$  ( $i$  — исходное состояние автомата;  $j, r$  — конечное состояние соответственно при отсутствии и наличии неисправности) называется техническое состояние объекта, при котором вместо перехода  $P_{ij}$  на заданном входном слове выполняется любой разрешенный для данного автомата переход  $P_{ir}$ .

Порождение кратных неисправностей представляется следующими вариантами:  $P_{ij}/P_{ir} + P_{ie}$ ;  $P_{ij}/\sum P_{ir}$  ( $r \neq j$ ), где  $\sum P_{ir}$  — произвольное подмножество разрешенных переходов;  $P_{ij}/P_{ij} + P_{ir}$ ;  $P_{ij}/P_{ij} + \sum P_{ir}$ . Ввиду наличия в конечном автомате значительного количества кратных дефектов их моделирование считается практически невозможным для оценки полноты тестов. В этом случае следует генерировать гарантированно полные относительно упомянутых неисправностей, 1999, № 2

равностей тесты, для чего следует теоретически доказать проверяющие способности проектируемых процедурой тестов. Однако такие доказательство не носят универсального характера, поэтому являются действительными для достаточно узкого класса цифровых объектов. Следовательно, рассмотрение и моделирование ограниченного класса дефектов — ОНП есть шаг вперед по пути создания универсальных средств оценки качества тестов и их генерации, хотя они, конечно, не покрывают все кратные неисправности переходов. В защиту ограниченной модели ОНП следует заметить, что общепринятая модель ОКН линий, покрывая более 90% всех дефектов, не может отвечать за отдельные кратные или функциональные неисправности. Например, тест для ПЭ 2И-НЕ (01, 10, 11), полный относительно ОКН, не выявляет функциональную неисправность, которая определяется таблицей истинности

$$C = \begin{Bmatrix} X_1 & X_2 & Y \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{Bmatrix},$$

где искажается состояние выхода на первом входном наборе. Тем не менее модель ОКН принимается исследователями за почти "золотой" сравнительный эталон при введении других моделей и классов дефектов для проектирования проверяющих тестов.

Считаем, что все состояния  $P = (P_1, \dots, P_i, \dots, P_n)$  и переходы  $P_{ij}$  ( $i=1, n; j=1, k$ ) конечного автомата наблюдаемы и различимы. Следовательно, пересечение двух любых состояний или переходов равно "пусто", что является условием проверки ОНП:  $P_{ij} \cap P_{ie} = \emptyset$  ( $j \neq e$ ).

Если автомат находится в исправном переходе  $P_{ij}$ , это означает, что неисправности  $P_{ie}$  ( $e=1, k; e \neq j$ ) проверяются тестом, который обеспечивает переход  $P_{ij}$ . Естественно, что любой другой тест, обеспечивающий переход  $P_{ie}$  ( $e \neq j$ ), проверяет неисправность  $P_{ij}$ . Следовательно, для проверки всех ОНП, исходящих из состояния  $P_i$ , необходимы два любых тестовых набора, которые обеспечивают переходы в два различных состояния автомата. Если автомат реализован по модели Мура, то для каждого состояния существует входной набор, сохраняющий упомянутое состояние. Включив сохраняющий входной вектор в тест, получаем условия построения теста для ОНП автомата: для получения полного, проверяющего все ОНП теста необходимо выполнить обход всех состояний автомата, побывав в каждом по одному разу и сохранив его; при этом необходимо вернуться в начальное состояние обхода.

Для триггера, изображенного на рис. 3, выполним моделирование теста относительно ОНП.

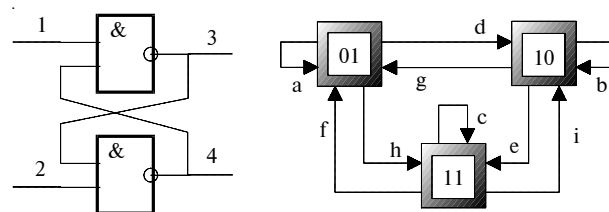


Рис. 3. Схема и граф переходов триггера

Триггер имеет три устойчивых состояния и девять устойчивых переходов, обозначенных буквами. И те, и другие различимы по наблюдаемым выходам. Моде-

лирование ОНП представлено в виде двоичного теста (Т) и соответствующих ему проверяемых ОНП (F):

$$[T, F] = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c & d & e & f & g & h & i \\ - & - & - & - & - & - & - & - & - \\ - & - & - & - & 1 & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 & - & - \\ - & - & - & - & - & 1 & - & - & 1 \\ - & - & 1 & - & - & - & - & - & 1 \\ - & - & - & 1 & - & - & - & - & 1 \\ - & - & - & - & 1 & - & - & - & 1 \\ 1 & - & - & - & - & - & - & 1 & - \end{bmatrix}$$

Поскольку неизвестно, в каком состоянии находился объект, на первом установочном входном наборе отсутствуют проверяемые дефекты. Все тестовые последовательности объединяются попарно, где первый вектор в паре является переключающим в новое состояние, второй — сохраняющим его. Последним следует вектор, переключающий автомат в начальное тестируемое состояние. Очевидно, для данной схемы тест, построенный по введенному правилу, является полным проверяющим и относительно ОКН. Однако это не всегда так. На рис. 4 изображены реверсивный счетчик и его граф пере-

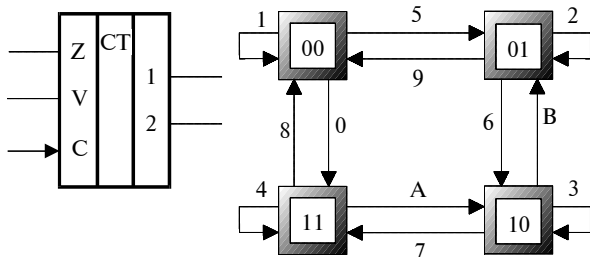


Рис. 4. Реверсивный счетчик

ходов, для которого полный проверяющий тест относительно ОНП

$$[T, F] = \begin{bmatrix} Z & V & C & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & A & B \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & 1 & - & - & - & - & 1 & - & - \\ 1 & - & - & - & - & - & - & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 & - & - & 1 & - & - \\ - & - & - & - & - & - & - & 1 & - & - & - & 1 \\ - & - & - & - & - & - & - & - & 1 & - & - & - \\ - & - & 1 & - & - & - & - & - & - & - & - & 1 \\ - & - & - & 1 & - & - & - & - & - & - & - & 1 \\ - & - & - & - & - & - & - & - & 1 & - & - & 1 \\ - & - & - & 1 & - & - & - & - & - & - & 1 & - \end{bmatrix}$$

не проверяет  $\neq 0$  на входе V, поскольку в режиме инкремента отсутствует его переключение. Однако более точная формулировка определяет отсутствие проверки режима декремента на предложенных тестовых наборах, что является минусом последних, которые и не были ориентированы на проверку каких-либо функций.

#### 4. Анализ КП УА для определения множества состояний

Приведенные рассуждения по поводу моделирования ОНП и построения тестов основываются на модели, изображенной графом переходов автомата. Его построение представляет собой задачу анализа исходной формы, в данном случае двухтактного кубического покрытия. Решением является множество устойчивых состояний, но не переходов, поскольку для построения полного относительно ОНП теста достаточно дважды побывать в каждом состоянии. Для моделирования всех неисправных переходов такое условие при оценке качества проверяющих последовательностей тоже приемлемо. Поэтому анализ КП сводится к определению

количества невыходных переменных, формирующих устойчивые состояния.

На примере синхронизированного задним фронтом JK-триггера, представленного кубическим покрытием: рассмотрим анализ его КП. Для корректно построенного КП символы SS в первом кубе свидетельствуют о четырех возможных состояниях переменных YQ, принимающих значения 00, 01, 10, 11. Анализ остальных кубов подтверждает сказанное, не исключая ни одно из упомянутых состояний в качестве запрещенного или недостижимого. Далее построение теста и оценка его полноты будут привязываться к полученным разрешенным устойчивым состояниям. Для JK-триггера не все переменные, определяющие состояния, являются наблюдаемыми. В этом и других аналогичных случаях необходимо обеспечить выполнение сканирования полученного состояния через наблюдаемые выходы заданием соответствующих входных условий. Для триггера значения ненаблюдаемой линии Y транспортируются через выход Q. Тест для JK-триггера, при условии непосредственного наблюдения линии Q для вывода информации с обеспечением сканирования переменной Y на наборах 2, 6, 10, имеет вид

$$[T, F] = \begin{bmatrix} J & C & K & Y & Q \\ 0 & 1 & 1 & 0 & X \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 00 & 01 & 10 & 11 \\ - & - & - & - \\ - & - & - & - \\ S & - & - & - \\ P & - & - & - \\ - & - & S & - \\ - & - & P & - \\ - & - & - & S \\ - & - & - & P \\ - & S & - & - \\ - & P & - & - \end{bmatrix}$$

Символы S и P на координатах проверяемых состояний обозначают их сохранение и переход в любое устойчивое на заданных входных наборах. Следовательно, для построения полного проверяющего все ОНП теста необходимо побывать во всех состояниях автомата, каждое из которых должно быть сохранено на одном входном наборе. Не учитывая установочные последовательности, длина теста для сильносвязного или циклического графа автомата, имеющего n вершин, равна 2n.

Более сложной является задача проектирования и анализа КП для ГСА (рис. 5), заданной исходной таблицей переходов

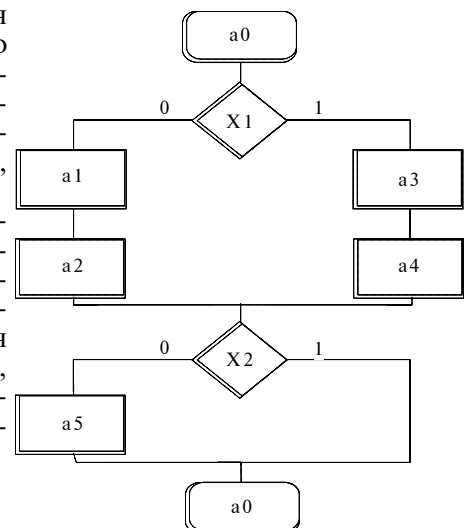


Рис. 5. Модель граф-схемы алгоритма

$$C = \begin{pmatrix} X_1 & X_2 & A_2^{t-1} & A_1^{t-1} & A_0^{t-1} & A_2^t & A_1^t & A_0^t \\ 0 & X & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & X & 0 & 0 & 0 & 0 & 1 & 1 \\ X & X & 0 & 0 & 1 & 0 & 1 & 0 \\ X & X & 0 & 1 & 1 & 1 & 0 & 0 \\ X & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ X & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ X & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ X & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ X & X & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Выполнив конкатенацию по соответствующим переменным  $a_i$  в моменты  $t-1$  и  $t$ , добавив переменные синхронизации и установки ( $C, Z$ ), а также кубы сохранения состояния и установки, получим двухтактное КП управляющего автомата, представленное покрытием

$$C = \begin{pmatrix} Z & C & X_1 & X_2 & A_2 & A_1 & A_0 \\ 0 & X & X & X & 0 & 0 & 0 \\ 1 & E & 0 & X & Q & Q & E \\ 1 & E & 1 & X & Q & E & E \\ 1 & E & X & X & Q & E & H \\ 1 & E & X & X & E & H & H \\ 1 & E & X & 0 & E & H & E \\ 1 & E & X & 1 & Q & H & Q \\ 1 & E & X & 0 & J & Q & E \\ 1 & E & X & 1 & H & Q & Q \\ 1 & E & X & X & H & Q & H \\ 1 & F & X & X & S & S & S \end{pmatrix}.$$

Анализ КП или таблицы переходов определяет шесть устойчивых состояний из восьми возможных, определяемых кубом 11 (110 и 111 не существуют как достижимые), относительно которых необходимо строить тест и моделировать неисправности. Конечный результат упомянутых действий представлен в виде

$$[T, F] = \begin{pmatrix} Z & C & X_1 & X_2 & A_2 & A_1 & A_0 & 000 & 001 & 010 & 011 & 100 & 101 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - & - & - \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & S & - & - & - & - & - \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & P & - & - & - & - & - \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & - & - & - & S & - & - \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & - & - & - & P & - & - \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & - & - & - & - & S & - \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & - & - & - & - & - & P \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & S & - & - & - & - & - \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & P & - & - & - & - & - \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & - & S & - & - & - & - \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & - & P & - & - & - & - \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & - & - & S & - & - & - \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & - & - & P & - & - & - \end{pmatrix}.$$

Если учесть, что  $S$  и  $P$  – символы двухтактного алфавита, обозначающие проверяемые неисправные переходы, которые в совокупности представляют символ  $Y = \{S, P\}$ , то буква  $Y$  может служить идентификатором полноты проверки ОНП на линиях, определяющих состояния конечного автомата.

## 5. Заключение

Для моделирования ОНП необходимо выполнить анализ КП или таблицы переходов, или графа переходов с целью определить множество достижимых устойчивых состояний. Для генерации минимального количества входных тестовых последовательностей необходимо иметь КП или иную форму описания УА. Тестовые наборы целесообразно упорядочить и разделить по триадам, в которых последовательно идущие две смежные пары соседних

векторов определяют сохранение состояния и переход в другое разрешенное. Первая проверяет одиночные неисправности переходов в другие состояния, идентифицируемые символом  $S$ , вторая – ОНП сохранения состояния как минимум, которые отмечаются символом  $P$ . Проверка дефектов типа  $S$  и  $P$  для текущего состояния обозначается символом  $Y$  и является полной. Проверка синхронизированного фронтом автомата заключается в подаче теста, обеспечивающего переходы во все возможные состояния с возвратом в начальное.

Если состояния не наблюдаемы, проектирование теста усложняется необходимостью транспортирования значений автоматных переменных к выходным линиям. Идеальным в этом смысле является тестопригодный проект ЦУ, выполненный по правилам SP, LSSD [9], где после установки автомата в очередное состояние можно задать режим сканирования – вывод значений триггеров, соединенных в сдвиговый регистр, на наблюдаемый выход.

Представлена модель тестирования цифровых синхронных автоматов, описанных на алгоритмическом уровне в форме граф-схем алгоритмов, при задании класса дефектов, определяемого одиночными неисправностями переходов, не модифицирующими совокупное и разрешенное пространство состояний и переходов.

Предложены процедура моделирования введенного класса дефектов и метод генерации тестов для них, основанный на выполнении двойного обхода графа состояний автомата, что является более экономичным по отношению к алгоритмам построения тестов, использующим структуру цифрового устройства.

**Литература:** 1. Хаханов В.И. Техническая диагностика элементов и узлов персональных компьютеров. К.: ІЗМН. 1997. 308с. 2. Breuer M.A., Friedman A.D. Diagnosis and reliable design of digital system. Woodlound wills, Computerscience press inc., 1976. 308p. 3. Автоматизированное проектирование цифровых устройств/ С.С.Бадудин, Ю.М.Барнаулов и др./ Под ред. С.С. Бадудина. М.: Радио и связь, 1981. 240с. 4. Автоматизация диагностирования электронных устройств/ Ю.В.Малышенко и др./ Под ред. В.П. Чипулиса. М.: Энергоатомиздат, 1986. 216с. 5. Airlau, Roland. Circuits Synthesis with VHDL. Kluwer Academic, 1994. – 240. 6. Ashenden, Peter. Designer's Guide to VHDL. Morgan Kaufman, 1994. 500p. 7. Thomas D.E. & Moorby, P. Verilog Hardware Description Language. Kluwer Academic, 1991. 223p. 8. Baker, Louis. VHDL Programming: with Advanced Topics. John Wiley & Sons, 1992. 384p. 9. Беннеттс Р.Д. Проектирование тестопригодных логических схем: Пер.с англ. Дербуновича Л.В. М.: Радио и связь, 1990. 176с.

Поступила в редколлегию 12.06.99

**Рецензент:** д-р техн. наук, проф. Кривуля Г.Ф.

**Хаханов Владимир Иванович**, д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей. Увлечения: баскетбол, горные лыжи. Адрес: Украина, 61726, Харьков, пр. Ленина, 14, тел. 40-93-26.

**Сысенко Ирина Юрьевна**, студентка факультета КИУ ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств. Хобби: иностранные языки, музыка. Адрес: Украина, 61726, Харьков, пр. Ленина, 14, тел. 40-93-26.

**Чамян Айка Левонович**, студент факультета КИУ ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств, сетей. Адрес: Украина, 61726, Харьков, пр. Ленина, 14, тел. 40-93-26.