

# Оптимизация, системный анализ и исследование операций

© 2021 г. А.Ю. МАТРОСОВА, д-р техн. наук (mau11@yandex.ru),  
С.В. ЧЕРНЫШОВ, (svchernyshov@mail.ru),  
О.Х. КИМ, (oh.kim@yandex.ru),  
Е.А. НИКОЛАЕВА, канд. техн. наук (nikolaeva-ea@yandex.ru)  
(Национальный исследовательский Томский государственный университет)

## ПОСТРОЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ, ОБНАРУЖИВАЮЩЕЙ РОБАСТНО ТЕСТИРУЕМЫЕ НЕИСПРАВНОСТИ ЗАДЕРЖЕК ПУТЕЙ В СХЕМАХ С ПАМЯТЬЮ

Предлагается метод построения последовательности булевых векторов входных переменных, доставляющей тестовые пары  $(v_1, v_2)$  соседних векторов в пространстве входных и внутренних переменных для робастно тестируемых неисправностей задержек путей (робастных Path Delay Faults (PDFs)) в логических схемах с памятью. Целью данной работы является выяснение возможности построения тестовой последовательности для заданного подмножества путей без использования технологий сканирования, т.е. без дополнительных аппаратных затрат в рамках ограничения на длину последовательности для отдельного пути. Проведенные эксперименты показывают, что тестовые последовательности удается построить не для всех путей (иногда ни для одного), для которых существуют тестовые пары в комбинационной составляющей схемы с памятью.

*Ключевые слова:* логическая схема с памятью, установочная последовательность, Reduced Ordered Binary Decision Diagram (ROBDD), робастно тестируемая неисправность задержки пути (PDF), rising (falling) transition.

DOI: 10.31857/S0005231021110106

### 1. Введение

Одним из основных параметров логических схем высокой производительности является высокая тактовая частота, т.е. высокая скорость функционирования схемы. Она определяется исходя из скорости прохождения сигналов через логические элементы вдоль путей, соединяющих входы и выходы схемы. Производители стремятся к увеличению тактовой частоты.

Для определения тактовой частоты схемы выделяют путь с максимальной задержкой — критический путь. Задержка этого пути определяет скорость функционирования схемы. Однако при высокой скорости функционирования и высоком уровне интеграции в схемах возникают непредусмотренные

разработчиками емкости, индуктивности, сопротивления, которые не удается рассчитать заранее. Это приводит к дополнительным задержкам в схеме и снижению ее расчетного быстродействия, что нежелательно. Модель неисправности задержки пути (Path Delay Fault (PDF) является одной из наиболее распространенных моделей задержек логической схемы, используемых на практике при тестировании логических схем. При использовании модели неисправности задержки пути предполагают, что задержки отдельных элементов пути и линий связи между ними малы, однако задержка пути в целом может превышать время между соседними синхросигналами тактовой частоты и искажать функционирование схемы. Выделяют робастно и не робастно тестируемые неисправности задержек путей. Под робастно тестируемой неисправностью понимают неисправность задержки пути, которая обнаруживается независимо от существования в схеме задержек других путей, превышающих допустимые значения. В противном случае неисправность называется не робастно тестируемой. При проявлении робастно тестируемой неисправности удается точно определить путь, на котором задержка имеет место, т.е. превышает время между соседними синхросигналами тактовой частоты. Проявление не робастно тестируемой неисправности такой возможности не дает. Информация о полученных задержках позволяет доработать схему с целью сохранения ее расчетного быстродействия или замаскировать действие обнаруженной задержки [1] с той же целью. Для тестирования задержек путей используются пары векторов  $(v_1, v_2)$ . Векторы каждой пары могут отличаться друг от друга по переменной, отмечающей начало рассматриваемого пути, а возможно, и значениями других переменных. Договоримся в дальнейшем векторы (булевы) и наборы тестовой пары использовать как синонимы. Задержки противоположных перепадов значений сигналов (rising transition и falling transition) могут отличаться, поэтому необходимо для каждого пути в общем случае иметь две тестовые пары. Под rising transition будем, как это принято в зарубежных источниках, понимать последовательность смены значений сигналов вдоль пути, заканчивающуюся на его выходе изменением нулевого сигнала на единичный. Соответственно, под falling transition — последовательность смены значений сигналов вдоль пути, заканчивающуюся на его выходе изменением единичного сигнала на нулевой сигнал. Пусть в момент времени  $t$  с приходом синхросигнала на схему поступает вектор  $v_1$  тестовой пары. При поступлении следующего синхросигнала на схему поступает вектор  $v_2$ . Если с приходом третьего синхросигнала на выходе схемы наблюдается сигнал, значение которого отличается от ожидаемого, то считаем, что в схеме имеет место задержка в ее путях.

На практике при тестировании задержек путей используют различные методы сканирования. При реализации этих методов в рамках Launch-on-Shift (LOS) технологии одним из векторов пары является тестовый набор для константной неисправности внутреннего полюса схемы, а второй вектор получается сдвигом первого. При использовании Launch-on-Capture (LOC) технологии второй вектор получается на основе реакции комбинационной составляющей схемы на первый вектор. Понятно, что при таких способах формирования пар векторов далеко не всегда удается получать тестовые пары для робаст-

но тестируемых неисправностей задержек путей. Обычно при использовании этих технологий удается обнаруживать около 20% таких неисправностей, в то время как использование точных методов [2], гарантирующих построение тестовой пары для робастно тестируемой неисправности, если она существует, позволяет обнаруживать от 40 до 90% и более таких неисправностей для заданного множества путей комбинационной составляющей схемы с памятью. Точные методы ориентированы на снижение потребляемой мощности. Проблема снижения потребляемой мощности при тестировании задержек исследуется во многих публикациях [3–7]. Другой проблемой при тестировании задержек являются большие аппаратурные затраты, связанные с доставкой тестов в рамках технологий сканирования. Как известно [8, 9], в технологиях сканирования используют в линиях обратных связей специальные триггеры, более сложные, чем D-триггеры. В рабочем режиме функционирования схемы они выполняют, как D-триггеры, задержку на один такт сигнала, сопоставляемого переменной состояния (внутренней переменной), а в режиме тестирования схемы образуют сдвиговый регистр, в который такт за тактом вносится составляющая тестового вектора, соответствующая внутренним переменным. Дополнительная аппаратура требуется также для разделения режимов функционирования и тестирования схемы с памятью. Точные методы могут быть применены в рамках Random Access Scan (RAS) технологий, реализация которых требует, к сожалению, больших аппаратурных затрат, чем использование LOS или LOC технологий. Итак, перечисленные способы тестирования задержек путей связаны с существенными аппаратурными затратами, которых хотелось бы избежать. Исследования, выполненные в данной работе, позволяют выяснить, всегда ли это возможно.

Имеется множество тестовых пар соседних булевых векторов (всех или некоторых), обнаруживающих робастно тестируемую неисправность задержки пути в комбинационной составляющей схемы с памятью. Требуется построить входную последовательность, доставляющую одну (любую) тестовую пару заданного множества из начального состояния  $q_0$  схемы с памятью в рамках ограничения на длину последовательности. Предлагаются алгоритмы построения таких последовательностей в условиях, когда 1) начало пути отмечено входной переменной и 2) когда начало пути отмечено переменной состояний (внутренней переменной) схемы. Если искомые последовательности удастся найти для каждого из путей заданного множества, то, совместив их, обнаружим задержки путей без дополнительных аппаратурных затрат.

Во втором разделе обсуждаются свойства пар наборов для робастно тестируемых неисправностей задержек путей. В третьем разделе описывается процедура получения Reduced Ordered Binary Decision Diagram (ROBDD-графа), представляющего множество всех тестовых пар соседних наборов для робастно тестируемых неисправностей задержек путей. В четвертом разделе приводятся алгоритмы построения последовательности, доставляющей из начального состояния тестовую пару для робастно тестируемой неисправности задержки пути в схеме с памятью. В пятом разделе обсуждаются результаты экспериментов.

## 2. Некоторые свойства тестовых пар наборов для робастно тестируемых неисправностей задержек путей

Имеем одновыходную схему  $C$  и соответствующую эквивалентную нормальную форму (ЭНФ). В [10] рассматривается задача построения пар тестовых наборов для робастно и не робастно тестируемых неисправностей задержек путей на основе анализа ЭНФ. Наборы  $v_2$  тестовых пар являются тестовыми наборами для константных неисправностей литер ЭНФ.

В случае rising transition тестируется 0-константная неисправность литеры ЭНФ, соответствующей пути  $\alpha$  ( $a_p$ -неисправность). В присутствии неисправности все вхождения литеры заменяются в ЭНФ константой 0. Тестовый набор, обнаруживающий эту неисправность, является набором  $v_2$  тестовой пары, который порождает смену сигнала на выходе схемы с нулевого на единичное значение.

В случае falling transition тестируется 1-константная неисправность литеры ЭНФ, соответствующей пути  $\alpha$  ( $b_p$ -неисправность). В присутствии неисправности все вхождения литеры заменяются в ЭНФ константой 1. Тестовый набор, обнаруживающий эту неисправность, является набором  $v_2$  тестовой пары, который порождает смену сигнала на выходе схемы с единичного на нулевое значение.

В [10] показано, что для тестовых пар, обнаруживающих робастно тестируемые неисправности задержек путей, выполняются следующие условия:

- 1) если  $v_2$  есть  $a_p$ -тестовый набор, то  $v_1$  есть  $b_p$ -тестовый набор тестовой пары и наоборот;
- 2) на наборах тестовой пары функция, реализуемая схемой  $C$ , принимает противоположные значения;
- 3) минимально покрывающий интервал  $u$  векторов  $v_1, v_2$  ортогонален всем конъюнкциям ЭНФ, не содержащим литеру, сопоставляемую пути  $\alpha$ ;
- 4) наборы  $(v_1, v_2)$  тестовой пары порождаются одной и той же непустой конъюнкцией.

При замене набора  $v_1$  набором  $v_2$  схема может попасть в промежуточное состояние, являющееся тестовым набором  $v_1$  для другого пути схемы. В результате можно определить задержку другого пути, а не рассматриваемого. Такая ситуация исключается при выполнении условия 3. Для проверки условия 3 в [10] предлагается воспользоваться ЭНФ схемы. Однако ЭНФ, как правило, оказывается громоздкой даже для небольших схем. Для соседних наборов тестовой пары  $(v_1, v_2)$  при переходе от  $v_1$  к  $v_2$  промежуточных состояний не возникает. Использование этого факта избавляет от необходимости анализировать ЭНФ.

*Теорема 1. Соседние наборы тестовой пары  $(v_1, v_2)$ , в которой  $v_2$  есть  $a_p$ -тестовый набор, а  $v_1$  —  $b_p$ -тестовый набор или наоборот, обнаруживают робастно тестируемую неисправность задержки пути для обоих перепадов значений сигналов этого пути (для rising transition и falling transition).*

*Теорема 2. Если существует тестовая пара наборов  $(b_p, a_p)$ , не являющихся соседними и обнаруживающих робастно тестируемую неисправность задержки пути  $\alpha$ , то существует тестовая пара соседних наборов, обнаруживающая эту задержку.*

Сам факт существования тестовой пары соседних наборов для робастно тестируемых неисправностей при существовании тестовой пары для несоседних наборов, возможно впервые, был отмечен как следствие теорем о свойствах ЭНФ при подстановке в нее одинаковых для соседних наборов констант [11]. От ЭНФ остается дизъюнкция из литер, сопоставляемых переменной, отмечающей начало пути. Конфигурация этих литер позволяет определить, какой тип задержки пути тестируется предъявленной парой соседних наборов. В теореме 2 подтверждается этот результат на основе исследования свойств конъюнкций ЭНФ (в них никакие переменные константами не заменяются), обеспечивающих существование тестовой пары для робастно тестируемых неисправностей задержек путей [10].

Из сказанного следует, что если рассматриваемый путь является робастно тестируемым, то для него существует тестовая пара, состоящая из соседних наборов.

Это также означает, что для установления существования тестовой пары для робастно тестируемой неисправности задержки пути достаточно ограничиться поиском тестовых пар, состоящих из соседних наборов. Предложенный в [12] алгоритм построения всех тестовых пар соседних наборов для обнаружения робастно тестируемых неисправностей задержек путей гарантирует нахождение в схеме всех робастно тестируемых путей.

Для построения множества всех тестовых пар соседних векторов для робастно тестируемых неисправностей рассматриваемого пути в [12] предлагается предварительно вычислять булеву разность (булеву производную) этого пути. Метод вычисления булевой разности и представления ее в виде ROBDD-графа  $R(D_{path})$  основан на выполнении операций над ROBDD-графами, извлекаемыми из фрагментов комбинационной схемы, характеризующихся полиномиальной сложностью.

### **3. Получение тестовых пар соседних наборов для робастно тестируемых неисправностей из ROBDD $R(D_{path})$**

Всевозможные тестовые наборы  $v_2$  для обоих перепадов значений сигналов представлены в виде ROBDD  $R(D_{path})$  [12]. Напомним, что в случае rising transition  $v_2$  является тестовым набором для  $a_p$ -неисправности литеры, отмечающей начало пути  $\alpha$ . Для falling transition  $v_2$  является тестовым набором для  $b_p$ -неисправности литеры, отмечающей начало пути  $\alpha$ . Приведем процедуру извлечения всех тестовых пар [12] для рассматриваемого пути. Пусть начало пути  $\alpha$  отмечено переменной  $x_i$  без инверсии.

$R_{rise} = R(D_{path}) \& x_i(R(D_{path}) \& \bar{x}_i)$  — ROBDD-граф, представляющий все наборы  $v_2$  для rising transition.

$R_{fall} = R(D_{path}) \& \bar{x}_i(R(D_{path}) \& x_i)$  — ROBDD-граф, представляющий все наборы  $v_2$  для falling transition.

Для каждого типа перепадов значений сигналов пути  $\alpha$  приведены две формулы, поскольку литера ЭНФ, сопоставляемая рассматриваемому пути, может иметь различные знаки инверсии.

Обозначим через  $R'_{rise}$  ROBDD-граф, полученный из  $R_{rise}$  удалением переменной  $x_i$ , а через  $R'_{fall}$  ROBDD-граф, полученный из  $R_{fall}$  удалением переменной  $x_i$ . В обоих случаях знаки переменной  $x_i$  не имеют значения.

Обозначим символом  $R_{rob}$  ROBDD-граф, представляющий тестовые пары соседних по переменной  $x_i$  наборов для робастно тестируемых неисправностей задержек пути  $\alpha$ .

$$R_{rob} = R'_{rise} \& R'_{fall}.$$

Из процедуры построения  $R_{rob}$  следует, что граф не содержит переменную  $x_i$ , отмечающую начало пути. Путь от корня графа  $R_{rob}$  до его 1-концевой вершины представляет конъюнкцию такую, что булев вектор в пространстве переменных  $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$  задает тестовую пару в пространстве  $n$  переменных для рассматриваемого пути. Один из векторов пары получается приписыванием переменной  $x_i$  значения 0, а другой — значения 1.

Если на векторе  $v_2$  и выходе одновыходной схемы (для многовыходной схемы на выходе, сопоставляемом пути  $\alpha$ ) достигается значение 1, то  $v_2$  — вектор для rising transition, если значение 0, то  $v_2$  — вектор для falling transition.

В последовательностной схеме пары соседних булевых векторов, заданных ROBDD-графом  $R_{rob}$ , сопоставляются полным состояниям, т.е. в векторах пары выделяются входные составляющие и составляющие по переменным состояниям.

Из тестовой пары формируются тройки векторов, обнаруживающие задержки инверсных перепадов сигналов рассматриваемого пути, либо  $v_2-v_1-v_2$ , либо  $v_1-v_2-v_1$ . Это значит, что при использовании тестовых пар из соседних булевых векторов имеется возможность обнаруживать противоположные перепады значений сигналов рассматриваемого пути в комбинационной составляющей тремя векторами вместо четырех.

Однако если переменные состояния в комбинационной составляющей не доступны, предлагается из начального состояния  $q_0$  последовательностной схемы доставлять по отдельности пары для каждой из последовательностей перепадов рассматриваемого пути. Дело здесь в том, что необходимо не только попасть в состояние, сопоставляемое началу последовательности из трех векторов, но далее оказаться в специальной цепочке переходов, порождаемой тройкой тестовых векторов, чем длиннее цепочка, тем ниже вероятность попадания в нее.

#### 4. Алгоритмы построения последовательности, обнаруживающей робастно тестируемую неисправность задержки пути в схеме с памятью

Приведем алгоритмы построения последовательности векторов входных переменных последовательностной схемы, доставляющей тестовую пару на

входы комбинационной составляющей схемы из начального состояния  $q_0$ . Напомним, что речь идет о соседних булевых векторах тестовой пары по переменной, отмечающей начало исследуемого пути. Эти алгоритмы формируют вместе с соответствующими тестовыми парами последовательности, обнаруживающие робастно тестируемые неисправности задержек путей, по одной входной последовательности для каждой последовательности перепадов значений сигналов рассматриваемого пути из заданного множества путей.

Пусть последовательностная схема получена из State Transition Graph (STG) описания, в котором выполнено кодирование состояний. STG-описание поведения дискретного устройства с памятью используется в зарубежных публикациях с прошлого века. Оно применяется в условиях, когда символы входного и выходного алфавита конечного автомата (абстрактной модели поведения устройства с памятью) уже закодированы разработчиком устройства. В STG-описании условие перехода из одного состояния в другое (с одновременным формированием выходного символа) представлено в виде тричного вектора в пространстве входных переменных и переменных состояний схемы с памятью. Это позволяет в общем случае компактно задавать множество условий этого перехода по сравнению с таблицами переходов-выходов, используемых в теории конечных автоматов. В табл. 1 приведен пример такого описания. Здесь множество состояний представлено символами  $\{1, 2, 3, 4\}$ .

Сопоставив состояниям булевы векторы в пространстве переменных  $z_1, z_2$  следующим образом:  $1 - \begin{smallmatrix} z_1 z_2 \\ 0 0 \end{smallmatrix}$ ,  $2 - \begin{smallmatrix} z_1 z_2 \\ 0 1 \end{smallmatrix}$ ,  $3 - \begin{smallmatrix} z_1 z_2 \\ 1 0 \end{smallmatrix}$ ,  $4 - \begin{smallmatrix} z_1 z_2 \\ 1 1 \end{smallmatrix}$ , получим таблицу, представляющую систему частичных булевых функций на множестве входных переменных и переменных состояний (табл. 2). В таблице функции переходов отмечены переменными  $z'_1, z'_2$ , а функции выходов переменными  $y_1, y_2$ , и т.д.

Эта таблица есть задание на синтез устройства. Для кодирования состояний использованы коды минимальной длины. При получении задания на синтез часто используют другие коды, например равновесные коды, коды Бергера и их модификации [13, 14] и др.

Необходимо принять во внимание следующие факты:

1) если в диаграмме переходов, соответствующей STG-описанию, отсутствуют петли, то невозможно построить последовательность, доставляющую тестовую пару для путей, начала которых отмечено входной переменной схемы;

2) если в результате кодирования состояний в полученном множестве кодов (в рабочей области, задаваемой STG-описанием) отсутствуют соседние векторы, то невозможно построить последовательность, доставляющую тестовую пару для путей, начала которых отмечены переменной состояния (внутренней переменной) в последовательностной схеме.

Итак, множество всех пар соседних булевых векторов, задающих полные состояния схемы и обнаруживающих робастно тестируемые неисправности задержек рассматриваемого пути в комбинационной части последовательностной схемы, представлено ROBDD-графом  $R_{rob}$ . В ROBDD  $R_{rob}$  сначала

**Таблица 1.** STG-описание поведения схемы с памятью

$x_1$	$x_2$	$x_3$	$q$	$q'$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$
0	–	–	1	1	0	0	0	1	0
–	0	–	1	1	0	0	0	1	0
1	1	–	1	2	1	0	0	1	0
–	–	0	2	2	0	0	1	1	0
–	–	1	2	3	1	0	1	1	0
1	0	–	3	3	0	1	0	0	0
0	–	–	3	4	1	1	0	0	0
–	1	–	3	4	1	1	0	0	0
–	–	0	4	4	0	1	0	0	1
–	–	1	4	1	1	1	0	0	1

**Таблица 2.** Описание поведения схемы с памятью после кодирования состояний

$x_1$	$x_2$	$x_3$	$z_1$	$z_2$	$z'_1$	$z'_2$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$
0	–	–	0	0	0	0	0	0	0	1	0
–	0	–	0	0	0	0	0	0	0	1	0
1	1	–	0	0	0	1	1	0	0	1	0
–	–	0	0	1	0	1	0	0	1	1	0
–	–	1	0	1	1	0	1	0	1	1	0
1	0	–	1	0	1	0	0	1	0	0	0
0	–	–	1	0	1	1	1	1	0	0	0
–	1	–	1	0	1	1	1	1	0	0	0
–	–	0	1	1	1	1	0	1	0	0	1
–	–	1	1	1	0	0	1	1	0	0	1

ла выполняется разложение Шеннона по внутренним, а затем по входным переменным. Это следует из метода его построения [12].

Имея в виду, что тестовые пары  $(v_1, v_2)$ , задаваемые  $R_{rob}$ , состоят из входной и внутренней составляющих, представим их как:  $v_1 = v_1^{in}, v_1^s; v_2 = v_2^{in}, v_2^s$ , т.е. тестовую пару далее будем задавать следующим образом:  $(v_1^{in}, v_1^s; v_2^{in}, v_2^s)$ .

*Рассматриваются следующие ситуации:*

а) соседние булевы векторы, извлекаемые из  $R_{rob}$ , отличаются по входной переменной;

б) соседние булевы векторы, извлекаемые из  $R_{rob}$ , отличаются по внутренней переменной.

**Случай а).**

Рассмотрим тестовую пару  $(v_1^{in}, v_1^s; v_2^{in}, v_2^s)$  векторов, которую необходимо подать на комбинационную составляющую последовательностной схемы для обнаружения задержки заданного пути  $\alpha$  в условиях отличия векторов тестовой пары (составляющих  $v_1^{in}, v_2^{in}$ ) по входной переменной  $x_i$  и при совпадении составляющих по переменным состояниям. Для простоты положим, что пере-

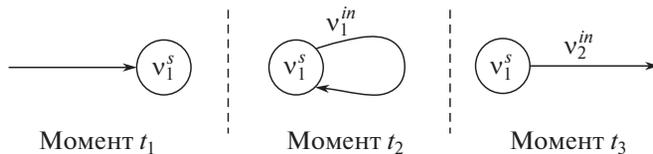


Рис. 1.

менная, отмечающая заданный путь, не имеет инверсии. Опишем процедуру доставки тестовой пары в соответствующие моменты времени:

1) предварительно с помощью подходящей входной последовательности, построение которой будет описано далее, устанавливаем схему (в момент  $t_1$ ) в состояние  $v_1^s$ . В данном случае не важен ни входной сигнал, ни состояние, из которого выполняется переход в состояние  $v_1^s$ ;

2) из состояния  $v_1^s$  под действием входного сигнала  $v_1^{in}$ , поступающего в момент  $t_2$  (именно в этот момент обеспечивается подача на входы комбинационной составляющей последовательной схемы первого вектора  $v_1^{in}$ ,  $v_1^s$  тестовой пары) переходим в состояние  $v_1^s$ , т.е. реализуем петлю в автоматной диаграмме переходов с целью достижения состояния  $v_1^s$  в момент  $t_3$ ;

3) в состоянии  $v_1^s$  в момент  $t_3$  подаем на входы схемы вектор  $v_2^{in}$ , т.е. обеспечиваем поступление второго вектора  $v_2^{in}$ ,  $v_1^s$  тестовой пары на входы комбинационной составляющей. Неважно, каким будет следующее состояние схемы, важно только, что должно произойти изменение сигнала на выходе, сопоставляемом рассматриваемому пути.

В момент  $t_4$  наблюдаем неисправность задержки на соответствующем пути выходе комбинационной составляющей, если неисправность задержки имеет место.

Заметим, что соседние векторы тестовой пары поступают в следующие друг за другом моменты времени. Переходы в моменты времени  $t_1, t_2, t_3$  иллюстрирует рис. 1.

Итак, для доставки тестовой пары необходимо из начального состояния  $q_0$  попасть в состояние  $v_1^s$ . Это можно сделать, построив последовательность, устанавливающую схему из начального состояния в некоторое состояние из множества состояний типа  $v_1^s$ . Причем представляют интерес не все состояния, имеющие петлю, а только те, в которых петля реализуется по входной составляющей  $v_1^{in}$ , содержащейся среди полных состояний, представленных  $R_{rob}$ . Обеспечив поступление в этом состоянии входной составляющей, далее формируем вектор  $v_2^{in}, v_1^s$ , заменив в  $v_1^{in}$  значение переменной  $x_i$  инверсным.

Для построения установочной последовательности из начального состояния  $q_0$  в состояние из заданного множества воспользуемся алгоритмом, предложенным в [15]. В нем множество состояний представлено ROBDD-графом  $R^{s_0}$ . Построение установочной последовательности из начального состояния является общей частью для различных типов перепадов сигналов

и различных типов переменных, отмечающих начало пути. Для каждой из этих ситуаций приходится строить свой граф  $R^{s_0}$ .

Итак, строим множество  $R^{s_0}$  для тестовой пары, в которой соседние векторы отличаются по входной переменной  $x_i$ . Из STG-описания выделяем фрагмент, представляющий петли в таблице переходов соответствующего конечного автомата. Фрагмент состоит из строк вида:

$$\begin{aligned}
 &k_1 q_i \rightarrow q_i \\
 &k_2 q_i \rightarrow q_i \\
 &\dots \\
 &\dots \\
 &k_m q_i \rightarrow q_i \\
 &k_{m+1} q_j \rightarrow q_j \\
 &\dots \\
 &\dots \\
 &k_{m+s} q_j \rightarrow q_j \\
 &\dots
 \end{aligned}$$

Здесь  $k_i$  — конъюнкции (троичные векторы) на множестве входных переменных схемы,  $q_i, q_j, \dots$  — состояния STG-описания (состояния конечного автомата), представленные булевыми векторами при кодировании. Левую часть рассматриваемого фрагмента, т.е. конъюнкции от входных переменных вместе с приписанными им состояниями представляем в виде ROBDD  $R_p$ . Этот граф пригодится при рассмотрении всех путей из предъявленного множества, начала которых отмечены входными переменными последовательностной схемы. Перейдем к рассмотрению заданного пути  $\alpha$ .

*Последовательность шагов для falling transition пути  $\alpha$ :*

1) из  $R_{rob}$  формируем  $a_p$ -тестовые наборы (наборы  $v_1^{in}, v_1^s$ ) тестовой пары, сопоставляемые входной литере  $x_i$ , представляя их соответствующим ROBDD-графом:  $R_{rob/x_i} = R_{rob} \& x_i$ ;

2) среди  $a_p$ -тестовых наборов выбираем такие, которые порождают петли:  $R_{a/p} = R_{rob/x_i} \& R_p$ , т.е. получаем множество всех векторов вида  $v_1^{in}, v_1^s$ , сопоставляемых моменту времени  $t_2$  при тестировании задержки пути;

3) выделяем в  $R_{a/p}$  множество внутренних состояний и представляем его графом  $R^{s_0}$ ;

4) получив вектор  $v_1^{in}, v_1^s$ , далее формируем вектор  $v_2^{in}, v_1^s$ , который порождается графом  $R_{rob}$ , и подаем его в момент  $t_3$ . Вектор  $v_2^{in}$  отличается от вектора  $v_1^{in}$  по переменной  $x_i$ .

При нахождении тестовой пары для rising transition выполняем следующие шаги алгоритма для  $b_p$ -тестовых наборов.

*Последовательность шагов для rising transition пути  $\alpha$ :*

1) из  $R_{rob}$  формируем  $b_p$ -тестовые наборы (наборы  $v_1^{in}, v_1^s$ ) тестовой пары, сопоставляемые входной литере  $\bar{x}_i$ , представляя их соответствующим ROBDD-графом:  $R_{rob/\bar{x}_i} = R_{rob} \& \bar{x}_i$ ;

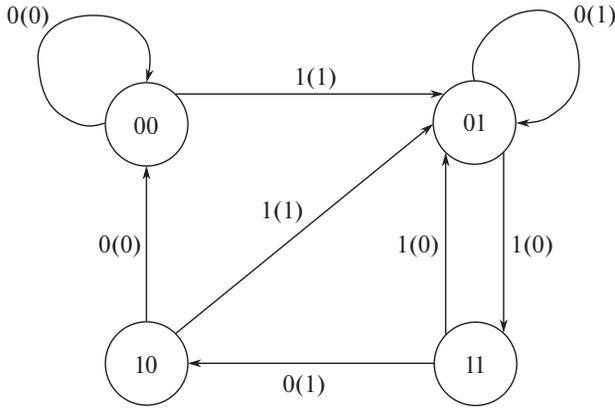


Рис. 2. Диаграмма переходов схемы с памятью.

2) среди  $b_p$ -тестовых наборов выбираем такие, которые порождают петли:  $R_{b/p} = R_{rob/\bar{x}_i} \& R_p$ , т.е. получаем множество всех векторов вида  $v_1^{in}, v_1^s$ , сопоставляемых моменту времени  $t_2$ , при тестировании задержки пути;

3) выделяем в  $R_{b/p}$  множество внутренних состояний и представляем его графом  $R^{s_0}$ ;

4) получив вектор  $v_1^{in}, v_1^s$ , далее формируем вектор  $v_2^{in}, v_1^s$ , который порождается графом  $R_{rob}$ , и подаем его в момент  $t_3$ . Вектор  $v_2^{in}$  отличается от вектора  $v_1^{in}$  по переменной  $x_i$ .

Рассмотрим иллюстрирующий пример. Пусть поведение автомата представляется диаграммой переходов, представленной на рис. 2. В диаграмме переходов состояния закодированы следующим образом: 1 —  $\overset{z_1 z_2}{00}$ , 2 —  $\overset{z_1 z_2}{01}$ , 3 —  $\overset{z_1 z_2}{10}$ , 4 —  $\overset{z_1 z_2}{11}$ . Имеем частный случай STG-описания, когда условия переходов представлены булевыми векторами. Положим, начальное состояние  $q_0$  представляется вектором  $\overset{z_1 z_2}{00}$ , сопоставляемым символу 1.

В диаграмме на дугах записаны входные и выходные (в скобках) векторы. В данном случае это однокомпонентные векторы. При кодировании состояний это описание превращается в задание системы булевых функций (см. табл. 3).

**Таблица 3.** Табличное задание системы булевых функций

$x$	$z_1$	$z_2$	$z'_1$	$z'_2$	$y$
0	0	0	0	0	0
0	0	1	0	1	1
1	0	0	0	1	1
0	1	0	0	0	0
1	1	0	0	1	1
0	1	1	1	0	1
1	1	1	0	1	0
1	0	1	1	1	0

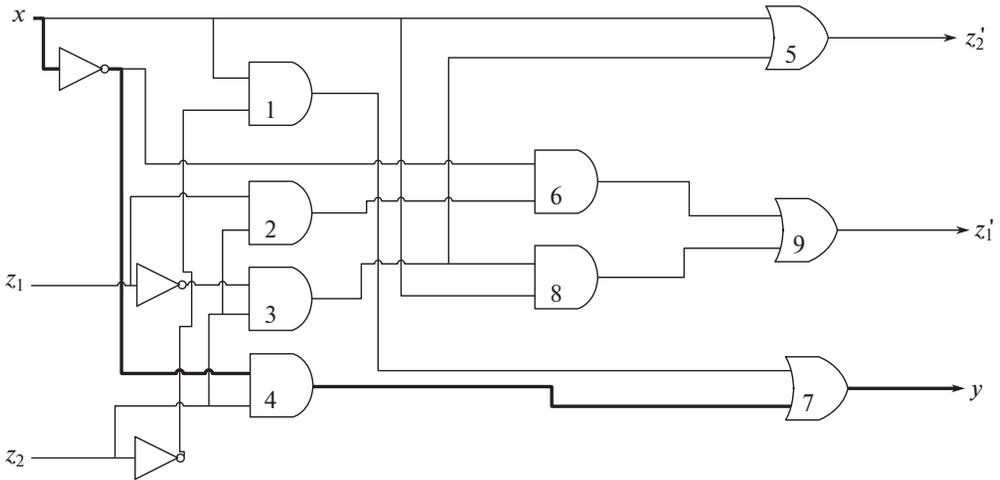


Рис. 3. Начало пути в схеме отмечено переменной  $x$ .

Из таблицы получаем реализацию системы булевых функций в виде системы ДНФ от входных переменных и переменных состояний:

$$\begin{aligned} z_1' &= \bar{x}z_1z_2 \vee x\bar{z}_1z_2; \\ z_2' &= x \vee \bar{z}_1z_2; \\ y &= x\bar{z}_2 \vee \bar{x}z_2. \end{aligned}$$

Эта система реализуется комбинационной составляющей схемы с памятью, показанной на рис. 3.

Рассмотрим путь  $x \rightarrow 4 \rightarrow 7 \rightarrow y$ .

Вычисляем для него булеву разность. В примере будем использовать операции над ДНФ, а не над ROBDD-графами для простоты. Чтобы сохранить соответствие тексту алгоритма, символ D (вместо ДНФ) будем сопровождать в скобках обозначением соответствующего ROBDD-графа из текста. Здесь символами  $U_1, U_2, \dots$  обозначены выходы соответствующих элементов схемы.

$$\begin{aligned} D_{U_7}/D_{U_4} &= (U_1 \vee U_4)|_{U_4=0} \oplus (U_1 \vee U_4)|_{U_4=1} = \bar{U}_1; \\ D_{U_4}/D_x &= (\bar{x}z_2)|_{x=0} \oplus (\bar{x}z_2)|_{x=1} = z_2; \\ D_{path} &= (\bar{x}\bar{z}_2)z_2 = (\bar{x} \vee z_2)z_2 = z_2. \end{aligned}$$

Итак,  $D_{path} = z_2$ .

Вычисляем далее  $D(R_{rob})$ .

$$\begin{aligned} D(R_{rise}) &= \bar{x}z_2; \\ D(R_{fall}) &= xz_2; \\ D(R'_{rise}) &= z_2; \\ D(R'_{fall}) &= z_2; \\ D(R_{rob}) &= z_2. \end{aligned}$$

Из табл. 3 выделяем переходы, сопоставляемые петлям в диаграмме переходов. Условия, обеспечивающие эти переходы, задаем соответствующей ДНФ.

$$D(R_p) = \overline{xz_1}\overline{z_2} \vee \overline{xz_1}z_2.$$

Для falling transition выполняем пп. 1–3 алгоритма и получаем состояние  $D(R^{s_0}) = \overset{z_1 z_2}{0} \overset{z_1 z_2}{1}$ :

1.  $D(R_{rob/\overline{x}}) = R_{rob}\overline{x} = \overline{x}z_2$ ;
2.  $D(R_{a/p}) = \overline{xz_1}z_2$ ;
3.  $D(R^{s_0}) = \overline{z_1}z_2$ .

Далее строим последовательность, доставляющую схему из начального состояния  $q_0(\overset{z_1 z_2}{0} \overset{z_1 z_2}{0})$  в состояние  $\overset{z_1 z_2}{0} \overset{z_1 z_2}{1}$ .

Пытаемся найти последовательность длины один, перемножая ДНФ, соответствующие переменным состояния:  $D(R^{s_1}) = xz_1 \vee x\overline{z_2} \vee \overline{xz_1}z_2$ .

Из полученной ДНФ следует, что из состояния  $\overset{z_1 z_2}{0} \overset{z_1 z_2}{0}$  под действием входного символа  $\overset{x}{1}$  попадаем в состояние  $\overset{z_1 z_2}{0} \overset{z_1 z_2}{1}$ . Из состояния  $\overset{z_1 z_2}{0} \overset{z_1 z_2}{1}$  под действием входного символа  $\overset{x}{0}$  оказываемся в том же состоянии  $\overset{z_1 z_2}{0} \overset{z_1 z_2}{1}$  (рис. 2). Это значит, что, воспользовавшись последовательностью длины один, в момент времени  $t_1$  попадаем в состояние, в котором можем организовать поступление нужной тестовой пары. А именно, под действием входного сигнала  $\overset{x}{0}$  формируем первый вектор тестовой пары в момент  $t_2$ . Оказавшись в следующий момент  $t_3$  в том же состоянии  $\overset{z_1 z_2}{0} \overset{z_1 z_2}{1}$ , подаем входной сигнал  $\overset{x}{1}$  и формируем второй вектор тестовой пары. Это можно увидеть из диаграммы переходов (рис. 2).

Для rising transition выполняем пп. 1, 2 алгоритма.

1.  $D(R_{rob/x}) = xz_2$ ;
2.  $D(R_{b/p}) = 0$ .

Поскольку  $D(R_{b/p}) = 0$ , приходим к заключению, что невозможно доставить тестовую пару для rising transition для рассматриваемого пути.

### Случай б).

Рассмотрим тестовую пару  $(v_1^{in}, v_1^s; v_1^{in}, v_2^s)$  векторов, которую необходимо подать на комбинационную составляющую последовательностной схемы для обнаружения задержки заданного пути  $\alpha$  в условиях отличия векторов тестовой пары (составляющих  $v_1^s, v_2^s$ ) по переменной состояния  $z_i$  и при совпадении составляющих по входным переменным. Для простоты положим, что переменная, отмечающая заданный путь, не имеет инверсии. Опишем процедуру доставки тестовой пары в соответствующие моменты времени:

1) предварительно с помощью подходящего входного воздействия устанавливаем схему (в момент  $t_1$ ) в состояние  $v_1^s$ . В данном случае не важен ни входной сигнал, ни состояние, из которого выполняется переход в состояние  $v_1^s$ ;

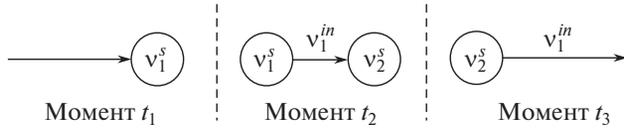


Рис. 4.

2) из состояния  $v_1^s$  под действием входного сигнала  $v_1^{in}$ , поступающего в момент  $t_2$  (именно в этот момент обеспечивается подача на входы комбинационной составляющей последовательностной схемы первого вектора  $v_1^{in}$ ,  $v_1^s$  тестовой пары), переходим в состояние  $v_2^s$ , т.е. переходим в соседнее состояние, отличающееся от  $v_1^s$  по переменной  $z_i$ ;

3) в состоянии  $v_2^s$  в момент  $t_3$  подаем на входы схемы вектор  $v_1^{in}$ , т.е. обеспечиваем поступление второго вектора  $v_1^{in}$ ,  $v_2^s$  тестовой пары на входы комбинационной составляющей. Неважно, каким будет следующее состояние схемы, важно только, что должно произойти изменение сигнала на выходе, сопоставляемом рассматриваемому пути.

В момент  $t_4$  наблюдаем неисправность задержки на соответствующем пути выходе комбинационной составляющей, если неисправность задержки имеет место.

Заметим, что соседние векторы тестовой пары поступают в следующие друг за другом моменты времени. Переходы в моменты времени  $t_1, t_2, t_3$  иллюстрируются рис. 4.

Итак, для доставки тестовой пары во всех отмеченных выше ситуациях необходимо из начального состояния  $q_0$  попасть в состояние  $v_1^s$ . Причем интерес представляют не все состояния, имеющие переходы в соседние состояния, а только те, в которых переход реализуется по входной составляющей  $v_1^{in}$ , содержащейся среди полных состояний, представленных  $R_{rob}$ . Обеспечив поступление в этом состоянии входной составляющей, далее формируем вектор  $v_1^{in}$ ,  $v_2^s$ , заменив в  $v_1^s$  значение переменной  $z_i$  на инверсное.

Из STG-описания выделяем фрагмент, представляющий переходы в соседние состояния. Фрагмент состоит из строк вида:

$$\begin{aligned}
 k_1 q_{i_1} &\rightarrow q_{j_1} \\
 k_2 q_{i_1} &\rightarrow q_{j_1} \\
 &\dots \\
 &\dots \\
 k_m q_{i_1} &\rightarrow q_{j_1} \\
 k_{m+1} q_{i_2} &\rightarrow q_{j_2} \\
 &\dots \\
 &\dots \\
 k_{m+s} q_{i_2} &\rightarrow q_{j_2} \\
 &\dots
 \end{aligned}$$

Здесь  $k_i$  – конъюнкции (троичные векторы) на множестве входных переменных схемы,  $q_{i_1}, q_{j_1}, q_{i_2}, q_{j_2} \dots$  – состояния STG-описания (состояния конечного автомата), представленные булевыми векторами при кодировании. Левую часть рассматриваемого фрагмента, т.е. конъюнкции от входных переменных вместе с приписанными им состояниями, представляем в виде ROBDD  $R_s$ . Этот граф пригодится при рассмотрении всех путей из предъявленного множества, начала которых отмечены переменными состояний последовательностной схемы. Перейдем к рассмотрению заданного пути  $\alpha$ .

*Последовательность шагов для falling transition пути  $\alpha$ :*

1) из  $R_{rob}$  формируем  $a_p$ -тестовые наборы (наборы  $v_1^{in}, v_1^s$ ) тестовой пары, сопоставляемые переменной состояний  $z_i$ , представляя их соответствующим ROBDD-графом:  $R_{rob/z_i} = R_{rob} \& z_i$ ;

2) среди  $a_p$ -тестовых наборов выбираем такие, которые порождают переходы в соседние состояния по переменной  $z_i$ :  $R_{a/s} = R_{rob/z_i} \& R_s$ , т.е. получаем множество всех векторов вида  $v_1^{in}, v_1^s$ , сопоставляемых моменту времени  $t_2$  при тестировании задержки пути;

3) выделяем в  $R_{a/s}$  множество внутренних состояний и представляем его графом  $R^{s_0}$ ;

4) получив вектор  $v_1^{in}, v_1^s$ , далее формируем вектор  $v_1^{in}, v_2^s$ , который порождается  $R_{rob}$ , и подаем его в момент  $t_3$ . Вектор  $v_2^s$  отличается от вектора  $v_1^s$  по переменной  $z_i$ .

При нахождении тестовой пары для rising transition выполняем следующие шаги алгоритма для  $b_p$ -тестовых наборов.

*Последовательность шагов для rising transition пути  $\alpha$ :*

1) из  $R_{rob}$  выбираем  $b_p$ -тестовые наборы (наборы  $v_1^{in}, v_1^s$ ) тестовой пары, сопоставляемые входной литере  $\bar{z}_i$ , представляя их соответствующим ROBDD-графом:  $R_{rob/\bar{z}_i} = R_{rob} \& \bar{z}_i$ ;

2) среди  $b_p$ -тестовых наборов выбираем такие, которые порождают переходы в соседние состояния по переменной  $z_i$ :  $R_{b/s} = R_{rob/\bar{z}_i} \& R_s$ , т.е. получаем множество всех векторов вида  $v_1^{in}, v_1^s$ , сопоставляемых моменту времени  $t_2$  при тестировании задержки пути;

3) выделяем в  $R_{b/s}$  множество внутренних состояний и представляем его графом  $R^{s_0}$ ;

4) получив вектор  $v_1^{in}, v_1^s$ , далее формируем вектор  $v_1^{in}, v_2^s$ , который порождается  $R_{rob}$ , и подаем его в момент  $t_3$ . Вектор  $v_2^s$  отличается от вектора  $v_1^s$  по переменной  $z_i$ .

Итак, во всех выше перечисленных алгоритмах (п. 3) для отыскания тестовых пар формируется соответствующее множество  $R^{s_0}$ .

Находим последовательность [15], устанавливающую схему из начального состояния  $q_0$  в одно из состояний соответствующего отыскиваемой паре множества  $R^{s_0}$ , т.е. в состояние  $v_1^s$ . Получив это состояние, используем для тестовой пары этого же типа ROBDD-граф из множества  $\{R_{a/p}, R_{a/s}, R_{b/p}, R_{b/s}\}$  и

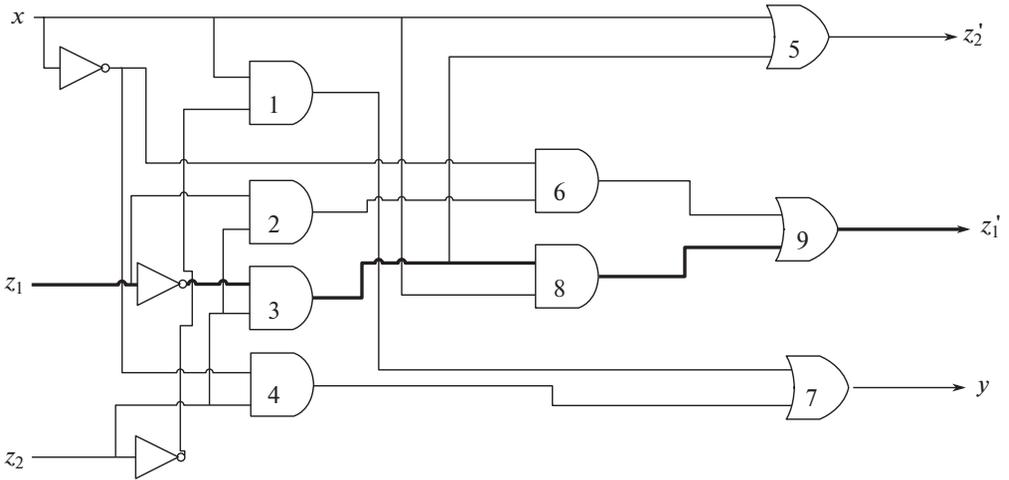


Рис. 5. Начало пути в схеме отмечено переменной  $z_1$ .

по нему находим первый вектор тестовой пары, а затем второй вектор способом, указанными в п.4 соответствующего отыскиваемой паре алгоритма.

Рассмотрим пример. В схеме задан путь  $z_1 \rightarrow 3 \rightarrow 8 \rightarrow 9 \rightarrow z'_1$  (см. рис. 5).

Начало пути отмечено переменной  $z_1$ .

Вычисляем для пути булеву разность.

$$\begin{aligned}
 D_{U_9}/D_{U_8} &= (U_8 \vee U_6)|_{U_8=0} \oplus (U_8 \vee U_6)|_{U_8=1} = U_6 \oplus 1 = \overline{U_6}; \\
 D_{U_8}/D_{U_3} &= (U_3x)|_{U_3=0} \oplus (U_3x)|_{U_3=1} = x; \\
 D_{U_3}/D_{z_1} &= (\overline{z_1}z_2)|_{z_1=0} \oplus (\overline{z_1}z_2)|_{z_1=1} = z_2; \\
 D_{path} &= (\overline{xz_1z_2})xz_2 = (x \vee \overline{z_1} \vee \overline{z_2})xz_2 = xz_2.
 \end{aligned}$$

Итак, булева разность представляется в виде  $D_{path} = xz_2$ .

Вычисляем далее  $D(R_{rob})$ .

$$\begin{aligned}
 D(R_{rise}) &= x\overline{z_1}z_2; \\
 D(R_{fall}) &= xz_1z_2; \\
 D(R'_{rise}) &= xz_2; \\
 D(R'_{fall}) &= xz_2; \\
 D(R_{rob}) &= xz_2.
 \end{aligned}$$

Получили единственную конъюнкцию, не содержащую переменную, отмечающую начало рассматриваемого пути. Конъюнкция порождает единственную тестовую пару. Получена  $\begin{matrix} xz_1z_2 & xz_1z_2 \\ (111, 101) \end{matrix}$ .

Из табл. 3 выделяем переходы, представляемые соседними векторами на множестве переменных  $z_1, z_2$ . Условия, обеспечивающие эти переходы, задаем в виде ДНФ.

$$D(R_s) = x\overline{z_1}\overline{z_2} \vee x\overline{z_1}z_2 \vee xz_1z_2 \vee \overline{x}z_1z_2 \vee \overline{x}z_1\overline{z_2}.$$

Рассматриваем falling transition. Выбираем вектор  $\overset{xz_1z_2}{101}$ . Это  $a_p$ -тестовый набор, поскольку он обращает в единицу ДНФ одновыходной подсхемы, которой принадлежит рассматриваемый путь.

Для falling transition выполняем пп. 1–3 алгоритма и получаем состояние  $D(R^{s_0}) = \overset{z_1z_2}{01}$ .

1.  $D(R_{rob/\bar{z}_1}) = x\bar{z}_1z_2$ ;
2.  $D(R_{a/s}) = x\bar{z}_1z_2$ ;
3.  $D(R^{s_0}) = \bar{z}_1z_2$ .

Для обеспечения доставки тестовой пары необходимо из начального состояния  $q_0$ , представляемого вектором  $\overset{z_1z_2}{00}$ , попасть в состояние  $\overset{z_1z_2}{01}$ . Ранее было показано, что в состоянии  $\overset{z_1z_2}{01}$  можно попасть из начального состояния  $\overset{z_1z_2}{00}$  по входному символу  $x$  ( $D(R^{s_1}) = xz_1 \vee x\bar{z}_2 \vee \bar{x}\bar{z}_1z_2$ ). В состоянии  $\overset{z_1z_2}{01}$  формируем тестовые пары  $\overset{xz_1z_2}{101}$  и  $\overset{xz_1z_2}{111}$ . Эту ситуацию можно видеть из диаграммы переходов (рис. 2).

Рассматриваем rising transition. Выбираем вектор  $\overset{xz_1z_2}{111}$ . Это  $b_p$ -тестовый набор, поскольку он обращает в ноль ДНФ одновыходной подсхемы, которой принадлежит рассматриваемый путь.

Пытаемся найти последовательность длины один, перемножая ДНФ, соответствующие переменным состояния:  $D(R^{s_1}) = x\bar{z}_1z_2$ . Полученная ДНФ в начальном состоянии  $\overset{z_1z_2}{00}$  не обращается в единицу. Следовательно, не существует последовательности длины 1, доставляющей нужную тестовую пару. Попробуем найти последовательность длины два. Это значит, что надо найти условия перехода из начального состояния в состояние  $\overset{z_1z_2}{01}$ . Ранее эти условия уже находили и представляли в виде ДНФ. Под действием входного символа  $x = 1$  из начального состояния  $\overset{z_1z_2}{00}$  попадаем в состояние  $\overset{z_1z_2}{01}$ . Далее из состояния  $\overset{z_1z_2}{01}$  под действием входного символа  $x$  попадаем в состояние  $\overset{z_1z_2}{11}$ . Итак, получили входную последовательность  $\overset{x}{1}, \overset{x}{1}$ , приводящую схему в состояние  $\overset{z_1z_2}{11}$ . Это можно видеть по диаграмме переходов (рис. 2). Формируем первый вектор тестовой пары  $\overset{xz_1z_2}{111}$ . Под действием входного сигнала  $x = 1$  переходим в состояние  $\overset{xz_1z_2}{101}$ , т.е. обеспечиваем поступление тестовой пары для rising transition. Эту ситуацию также можно видеть из диаграммы переходов (рис. 2).

Строим для каждого пути из заданного множества и типа перепадов значений сигналов соответствующие последовательности, обнаруживающие задержку пути, объединяем их и получаем тестовую последовательность для заданного множества путей. Отметим, что полученная тестовая последовательность состоит из фрагментов, начинающихся в состоянии  $q_0$ .

## 5. Результаты экспериментов

Описанные выше алгоритмы были реализованы в виде программы. Для операций над ROBDD-графами использовалась библиотека CUDD. Были проведены эксперименты на некоторых бенчмарках, результаты которых приведены в табл. 4. В схемах, представленных в бенчмарках, рассматривались только самые длинные пути. Длина установочной последовательности ограничивалась тысячей входными векторами, однако в процессе эксперимента установочные последовательности не достигали и сотни входных векторов. В последнем столбце таблицы показана доля (в процентах) путей, для которых из начального состояния последовательностной схемы удалось доставить тестовую пару, обнаруживающую робастно тестируемую неисправность задержки пути. Эта доля определяется по отношению ко всем путям, для которых существуют тестовые пары для тех же неисправностей в условиях доступности входов, сопоставляемых переменным состояниям комбинационной составляющей.

Результаты экспериментов, проведенных на контрольных схемах, показали, что доля путей, для которых возможна доставка тестовых пар, обнаруживающих робастно тестируемые неисправности задержек путей, в лучшем случае составляет сорок с небольшим процентов. Из девяти схем для двух вообще не удалось доставить тестовые пары ни для одного из рассмотренных путей. Это значит, что отказываться от технологий сканирования, несмотря на связанные с ними большие аппаратурные затраты, нецелесообразно.

Обозначения:

- name – название бенчмарка;
- i – количество входов;
- s – количество переменных состояний;
- xr – количество путей, начало которых помечено входной переменной и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для rising transition;
- xf – количество путей, начало которых помечено входной переменной и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для falling transition;

**Таблица 4.** Результаты экспериментов

name	i	s	xf	xr	zf	zr	xob	zob	ob	N	%
s27	4	3	8	8	18	2	8	2	10	58	17,2%
s208	11	8	0	0	0	3	0	0	0	643	0%
s298	3	14	40	0	0	0	0	0	0	1179	0%
s386	7	6	45	57	320	572	0	160	160	1351	11,8%
s510	19	6	0	0	906	724	0	575	575	1373	41,9%
s820	18	5	34	451	1860	1392	0	1313	1313	3055	42,87%
s832	18	5	34	450	1859	1391	0	1312	1312	3052	42,98%
s1488	8	6	0	58	2578	2209	0	1612	1612	5384	29,94%
s1494	8	6	0	58	2628	2247	0	1626	1626	5351	30,39%

—  $zr$  – количество путей, начало которых помечено переменной состояния и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для rising transition;

—  $zf$  – количество путей, начало которых помечено переменной состояния и для которых можно доставить тестовую пару из начального состояния схемы. Тестовая пара предназначена для falling transition;

—  $xob$  – количество путей, начало которых помечено входной переменной и для которых можно доставить тестовые пары для обоих перепадов значений сигналов из начального состояния схемы;

—  $zob$  – количество путей, начало которых помечено переменной состояний и для которых можно доставить тестовые пары для обоих перепадов значений сигналов из начального состояния схемы;

—  $ob$  – общее количество робастно тестируемых путей, для каждого из которых обнаружимы оба перепада значений сигналов;

—  $N$  – количество путей, для которых существуют непустые графы  $R_{rob}$ ;

—  $\%$  – доля (в процентах) робастно тестируемых путей, для которых удалось доставить тестовую пару, среди путей, для которых существуют непустые графы  $R_{rob}$ .

## 6. Заключение

Ранее был предложен метод отыскания всех тестовых пар, состоящих из соседних наборов булевых векторов, обнаруживающих робастно тестируемые неисправности задержек пути в условиях их доставки на входы комбинационной составляющей схемы с памятью. В данной работе предложен способ доставки таких тестовых пар из начального состояния  $q_0$  схемы с памятью в условиях ограничения на длину последовательности, основанный на операциях над ROBDD-графами. Работа ориентирована на исследование возможности отказа от использования техник сканирования, требующих больших аппаратных затрат. В работе выделены классы автоматов и способы кодирования состояний, для которых невозможна доставка тестовых пар, существующих в комбинационной составляющей. Проведенные эксперименты на контрольных примерах (benchmarks) показали, что далеко не всегда удается доставить хотя бы одну тестовую пару, не важно какую именно, для пути, имеющего тестовые пары для комбинационной составляющей в условиях доступности ее переменных состояния. Это значит, что для некоторых схем с памятью обнаружение робастно тестируемых неисправностей задержек путей без существенных аппаратных затрат может оказаться невозможным.

## ПРИЛОЖЕНИЕ

*Доказательство теоремы 1.* Будем иметь в виду, что  $a_p$ -тестовый набор тестовой пары для робастно тестируемой неисправности задержки пути обращает в единицу в общем случае несколько непустых конъюнкций ЭНФ (в непустых конъюнкциях ЭНФ отсутствуют взаимно инверсные переменные). Каждая из таких конъюнкций содержит литерал ЭНФ, сопоставляемый пути  $\alpha$  (литерал ЭНФ — это символ переменной с подходящим знаком

инверсии и последовательностью индексов, представляющих путь  $\alpha$ ). Сопоставляемая литералу переменная присутствует в такой конъюнкции только один раз. Это объясняется тем, что  $b_p$ -тестовый набор тестовой пары  $(b_p, a_p)$  порождается конъюнкцией, которая не содержит литералов, отличающихся от литерала, сопоставляемого пути  $\alpha$  только последовательностью индексов, сопоставляемых другому пути схемы. В то же время оба тестовых набора рассматриваемой пары порождаются одной и той же непустой конъюнкцией ЭНФ (условие 4). Отметим, что тестовый набор для  $a_p$ -неисправности (по построению) ортогонален каждой конъюнкции ЭНФ, не содержащей литерала, сопоставляемого пути  $\alpha$ . Тестовый набор для  $b_p$ -неисправности (по построению) ортогонален всем конъюнкциям ЭНФ. Примем во внимание, что любой тестовый набор является булевым вектором и, следовательно, он ортогонален пустой конъюнкции. Это значит, что минимально покрывающий интервал  $u$ , порожденный соседними векторами  $(b_p, a_p)$ , ортогонален каждой конъюнкции ЭНФ, не содержащей литерал, сопоставляемый пути  $\alpha$ . Итак, для рассматриваемых соседних наборов выполняются условия быть тестовой парой для робастно тестируемой неисправности задержки пути, перечисленные ранее в работе. Имея в виду теорему 5 работы [7], приходим к заключению, что найденная тестовая пара может быть использована для обнаружения противоположных перепадов значений сигналов рассматриваемого пути. Теорема доказана.

*Доказательство теоремы 2.* Пусть  $u$  — минимально покрывающий интервал наборов  $(b_p, a_p)$  тестовой пары для робастно тестируемой неисправности задержки пути. Рассмотрим тестовый набор  $a_p$ . Этот набор (по построению) ортогонален каждой конъюнкции, не содержащей литерал, сопоставляемой пути  $\alpha$ , и, возможно, пересекается с некоторыми остальными конъюнкциями рассматриваемой ЭНФ. Построим для  $a_p$  соседний по переменной  $x_i$  набор  $b'$ . Здесь  $x_i$  — переменная, отмечающая начало пути, для которого построена тестовая пара  $(b_p, a_p)$ . Набор  $b'$  поглощается интервалом  $u$ , следовательно, тоже ортогонален каждой конъюнкции, не содержащей литерал, сопоставляемой пути  $\alpha$ . Кроме того,  $b'$  ортогонален всем конъюнкциям, содержащим литерал, сопоставляемый пути  $\alpha$ , по переменной  $x_i$ , т.е.  $b'$  ортогонален конъюнкциям ЭНФ в целом. Это значит, что он принадлежит области нулевых значений функции, представляемой ЭНФ, и является  $b_p$ -тестовым набором. Теорема доказана.

## СПИСОК ЛИТЕРАТУРЫ

1. *Matrosova A., Ostanin S., Chernyshov S.* Masking Robust Testable PDFs // Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS), 13–16 september 2019, Batumi. Kharkov: IEEE, 2019. P. 420–423.
2. *Matrosova A. Yu., Andreeva V. V., Tyshinskiy V. Z., Goshin G. G.* Applying ROBDDs for delay testing of logical circuits. // Russ. Physics J. 2019. V. 62. No. 5. P. 615–621.
3. *Lindgren P., Kerttu M., Thornton M., Drechsler R.* Low power optimization technique for BDD mapped circuits // In Proceedings of the 2001 Asia and South Pacific Design Automation Conference (ASP-DAC '01), Association for Computing Machinery, New York, NY, USA, P. 615–621.

4. *Shelar R.S., Sapatnekar S.S.* An efficient algorithm for low power pass transistor logic synthesis // Proceedings of ASP-DAC/VLSI Design 2002, 7th Asia and South Pacific Design Automation Conference and 15th International Conference on VLSI Design, Bangalore, India, 2002, P. 87–92.
5. *Gekas G., Nikolos D., Kalligeros E., Kavousianos X.* “Power aware test-data compression for scan-based testing”, ICECS 2005 // 12th IEEE International Conference on Electronics, Circuits and Systems, Gammarrth, Tunisia, 2005, P. 1–4.
6. *Tudu J.T., Larsson E., Singh V., Agrawal V.D.* On Minimization of Peak Power for Scan Circuit during Test // Test Symposium 2009 14th IEEE European, 2009, P. 25–30.
7. *Kotasek Z., Skarvada J., Strnadel J.* Reduction of Power Dissipation Through Parallel Optimization of Test Vector and Scan Register Sequences // IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems, 2010, P. 364–369.
8. *Eichelberger E.B., Williams T.W.* A Logic Design Structure for LSI // Proc. 14th Design Automation Conference. 1977. P. 462–468.
9. *Agrawal V.D., Cheng K.T., Johnson D.D.* Designing Circuits with Partial Scan // IEEE Design and Test of Computers. 1988. P. 8–15.
10. *Матросова А.Ю., Липский В.Б.* Свойства пар тестовых наборов, обнаруживающих неисправности задержек путей в логических схемах VLSI высокой производительности // АИТ. 2015. № 4. С. 135–148.  
*Matrosova A. Yu., Lipskii V.B.* Properties of pairs of test vectors detecting path delay faults in high performance VLSI logical circuits // Autom. Remote Control. 2015. No. 4. P. 135–148.
11. *Сапожников В.В., Сапожников Вл.В., Лыков А.А.* Теоремы анализа для обнаружения неисправности типа “временная задержка” // Электрон. моделирование. 2004. Т. 266. No. 3. С. 83–93.
12. *Matrosova A. Yu., Andreeva V. V., Nikolaeva E. A.* Finding Test Pairs for PDFs in Logic Circuits Based on Using Operations on ROBDDs // Russ. Physics J. 2018. V. 61. No. 5. P. 994–999.
13. *Efanov D., Sapozhnikov V., Sapozhnikov Vl., Nikitin D.* Sum Code Formation with Minimum Total Number of Undetectable Errors in Data Vectors // Proceedings of 13th IEEE East-West Design & Test Symposium (EWDTS’2015), Batumi, Georgia, September 26–29, 2015, P. 141–148.
14. *Efanov D. V., Sapozhnikov V., Sapozhnikov Vl.* Two-Modulus Codes with Summation of One-Data Bits for Technical Diagnostics of Discrete Systems // Automatic Control and Computer Sciences. 2018. V. 52. Issue 1. P. 5–21.
15. *Matrosova A., Andreeva V., Melnikov A.* ROBDDs Application for Finding the Shortest Transfer Sequence of Sequential Circuit or Only Revealing Existence of this Sequence without Deriving the Sequence itself // Proceedings of IEEE East-West Design & Test Symposium (EWDTS’2016). Yerevan: IEEE Computer Society, 2016. P. 513–516.

*Статья представлена к публикации членом редколлегии М.Ф. Караваем.*

Поступила в редакцию 22.05.2020

После доработки 01.02.2021

Принята к публикации 16.03.2021