

Построение тестовых последовательностей для робастно тестируемых неисправностей задержек путей с низкой потребляемой мощностью с использованием SAT-решателей и ROBDD-графов

А.Ю. Матросова, В.З. Тычинский, В.В. Андреева

Институт прикладной математики и компьютерных наук, Национальный исследовательский
Томский государственный университет, г. Томск

mau11@yandex.ru, tvz.041@gmail.com, avv.21@mail.ru

Аннотация — Предлагается новый подход к построению тестовых последовательностей, обнаруживающий робастно тестируемые неисправности задержек путей (Path Delay Faults (PDFs)), ориентированный на снижение потребляемой мощности при тестировании. Снижение потребляемой мощности обеспечивается сокращением числа перепадов значений сигналов в тестовой последовательности и сокращением ее длины. Предлагаемый подход основан на получении и компактном представлении множества всех тестовых пар соседних булевых векторов для пути в логической схеме в виде ROBDD-графа. Каждая пара порождает три соседних булевых вектора, обнаруживающих робастно тестируемую неисправность задержки пути (robust testable Path-Delay Faults) для обоих перепадов значений сигналов пути. Соседними называются булевы векторы, отличающиеся значениями только одной компоненты. С целью построения как можно более длинных фрагментов (больше трех векторов) тестовых последовательностей соседних векторов, то есть фрагментов с минимальным потреблением мощности, выполняется пересечение графов рассматриваемых путей. Разработано несколько алгоритмов формирования тестовых последовательностей на основе получения и совмещения таких фрагментов. Традиционные подходы к тестированию robust testable PDFs ориентированы на нахождение хотя бы одной тестовой пары для каждого из перепадов значений сигналов, в этих условиях построение вышеупомянутых фрагментов длины больше двух оказывается практически невозможным. В предлагаемом нами подходе множества тестовых пар строятся либо с использованием операций над ROBDD-графами, либо с совместным применением этих операций и SAT-решателей. Проведенные эксперименты демонстрируют высокое качество тестовых последовательностей в смысле потребляемой мощности, не достижимое традиционными методами, в том числе для схем, поведение которых не удается представить ROBDD-графом в силу его большой размерности.

Ключевые слова — робастные (робастно тестируемые) неисправности задержек путей (robust testable PDFs), reduced ordered binary decision diagrams (ROBDD-графы), SAT-решатели (SAT solvers), комбинационные схемы, последовательностные схемы.

I. ВВЕДЕНИЕ

Известно, что тестирование логических схем, как правило, сопровождается потреблением большей мощности, чем работа схемы в режиме ее нормального функционирования. В результате схема может быть разрушена в процессе тестирования, в то время как она могла бы работать в нормальном режиме функционирования. В связи с этим разработка методов тестирования, обеспечивающих снижение потребляемой мощности, является актуальной проблемой для интегральных схем высокой производительности. В таких схемах в процессе их создания обязательно выполняется тестирование неисправностей задержек путей. На практике применяется метод подачи тестовой пары в режиме сканирования, причем, один из наборов тестовой пары является тестовым набором для константных неисправностей на полюсах вентилях, а другой получается либо сдвигом первого набора, либо строится на основе реакции комбинационной составляющей схемы с памятью на первый набор. Такой (эвристический) подход к построению тестовых пар обеспечивает обнаружение около 20% робастно тестируемых неисправностей задержек путей. Обнаружение робастно тестируемых неисправностей позволяет точно определить путь в схеме, на котором задержка проявляется, в то время как обнаружение не робастно тестируемой неисправности такой возможности не дает. Обычно информация об обнаруженных робастно тестируемых неисправностях выдается инженерам с тем, чтобы они попытались скорректировать схему, с целью сохранения ее расчётного быстродействия. Возможно также маскирование обнаруженных неисправностей задержек путей [1]. Мы использовали метод [2] нахождения тестовых пар, гарантированно обнаруживающий робастно тестируемую неисправность задержки пути и компактно представляющий множество всех тестовых пар соседних булевых векторов в виде ROBDD-графа R_{rob} . Проведенные на его основе эксперименты показали, что для большинства исследуемых схем процент робастно тестируемых путей лежит в пределах

от 40% до 90% и выше [3]. Каждая тестовая пара, представленная R_{rob} , порождает три соседних булевых вектора: либо (v_1, v_2, v_1) , либо (v_2, v_1, v_2) , которые обнаруживают робастно тестируемые задержки пути для обоих перепадов значений сигналов. В работе [4] предложен метод построения тестовых последовательностей, существенно сокращающий потребление мощности при использовании фрагментов из соседних векторов, то есть фрагментов с минимальным потреблением мощности за счет перепадов сигналов, и сокращения длины фрагментов при сохранении числа путей, для которых тестовая последовательность построена. Соединение фрагментов [4] также ориентировано на сокращение числа перепадов сигналов между крайними векторами соседних фрагментов. Эти результаты получены исключительно с применением операций над ROBDD графами, построенными для подсхем исследуемой логической схемы. Как известно, операции над ROBDD-графами характеризуются полиномиальной сложностью. Однако не для всех комбинационных схем (комбинационных составляющих схем с памятью) можно построить ROBDD-граф. Существуют схемы, для которых число вершин графа экспоненциально зависит от числа входов схемы при любом порядке разложения по переменным. Это значит, что для схем с несколькими десятками входов далеко не всегда удается построить ROBDD-граф.

Для таких схем в данной работе предлагается воспользоваться SAT решателями для извлечения тестовых пар рассматриваемого пути в виде безыбыточной ДНФ (БДНФ). Из полученных БДНФ различных путей схемы далее извлекаются ROBDD-графы. ROBDD-графы затем используются при построении тестовых последовательностей для заданного множества путей с применением пересечений графов различных путей [4]. Это обусловлено тем, что пересечение ROBDD-графов выполняется, как правило, быстрее, чем пересечение соответствующих графов БДНФ. Рассматриваемый в работе подход к построению тестовой последовательности ориентирован на Random Access Scan (RAS) технологии и использование операций над ROBDD-графами, там, где это возможно.

Заметим, что технологии сканирования связаны с большими дополнительными аппаратными затратами. Проведенные нами исследования в области построения тестовых последовательностей без привлечения дополнительных аппаратных затрат показали, что обеспечить доставку тестовой пары для робастно тестируемой неисправности задержки пути удается далеко не для каждого пути (иногда ни для одного), для которого существуют тестовые пары в комбинационной составляющей схемы с памятью. Возможность доставки тестовой пары из начального состояния схемы зависит от таблицы переходов соответствующего схеме автомата и способов кодирования его состояний. Это значит, что избежать дополнительных аппаратных затрат при

тестировании задержек в схемах с памятью пока что не удается.

В разделе II обсуждается проблема построения ROBDD-графа R_{rob} , представляющего всевозможные пары тестовых наборов для робастно тестируемых PDFs рассматриваемого пути. В разделе III обсуждаются различные алгоритмы построения тестовых последовательностей, основанные на пересечении ROBDD-графов и ориентированные на снижение потребляемой мощности. Результаты экспериментов приводятся в разделе IV.

II. ПОСТРОЕНИЕ ROBDD-ГРАФА R_{rob}

A. Вычисление булевой разности пути

Путём в комбинационной схеме называют последовательность элементов, в которой выход предыдущего элемента является входом последующего элемента. Один из входов первого элемента последовательности является входом схемы. Выход последнего элемента последовательности является выходом схемы.

Для тестирования задержки пути необходимы два вектора, тестовая пара (v_1, v_2) . Векторы пары отличаются инверсными значениями по переменной, отмечающей начало пути, а, возможно, и значениями по другим переменным. Задержки противоположных перепадов значений сигналов (rising transitions и falling transitions) могут различаться, и поэтому для каждого пути в общем случае строится две пары.

В работе [2] предлагается метод построения ROBDD-графа R_{rob} на основе вычисления булевой разности для заданного пути. Представим путь α одно выходной комбинационной схемы C , входы которой обозначены переменными x_1, \dots, x_n , последовательностью следующих символов: $x_i, u_1, u_2, \dots, u_{r-1}, u_r$. Здесь r – длина пути α , x_i – переменная, отмечающая начало пути (вход схемы C), переменные $u_1, u_2, \dots, u_{r-1}, u_r$ отмечают выходы элементов пути. Переменная u_r отмечает выход схемы C . Заметим, что схема C может быть комбинационной частью последовательностной схемы.

Пусть переменные u_i, u_{i-1} отмечают выходы соседних элементов пути α . Рассмотрим подсхему C_{u_i} схемы C . Выход этой подсхемы отмечается переменной u_i , а входы – переменными x_1, \dots, x_n, u_{i-1} . Здесь переменная u_{i-1} является входной переменной подсхемы C_{u_i} наряду с переменными x_1, \dots, x_n и одновременно входной переменной элемента с выходом, отмеченным переменной u_i .

Обозначим символом $D_{u_i} / D_{u_{i-1}}$ булеву разность, вычисляемую для функции, реализуемой подсхемой C_{u_i} , по переменной u_{i-1} . Тогда булева разность для пути α примет следующий вид:

$$D_\alpha = (D_{u_r} / D_{u_{r-1}}) \wedge (D_{u_{r-1}} / D_{u_{r-2}}) \wedge \dots \wedge (D_{u_2} / D_{u_1}) \wedge (D_{u_1} / D_{x_i}).$$

В. Получение тестовых наборов для робастно тестируемых неисправностей с использованием ДНФ

Выражение для булевой разности $D_{u_i} / D_{u_{i-1}}$ с использованием ДНФ имеет следующий вид: $D_{u_i} / D_{u_{i-1}} = f_{u_i}^{u_{i-1}=0} \oplus f_{u_i}^{u_{i-1}=1}$, где f_{u_i} представляет функцию подсхемы C_{u_i} и зависит от переменных x_1, \dots, x_n, u_{i-1} .

ДНФ D_α представляет все тестовые наборы v_2 для обоих перепадов значений сигналов (rising and falling transitions), которые не отделены друг от друга. Необходимо их разделить. ДНФ для rising transition обозначим через $D_{rise} = D_\alpha \wedge x_i$. ДНФ для falling transition обозначим через $D_{fall} = D_\alpha \wedge \bar{x}_i$.

Поскольку литеры ЭНФ, сопоставляемая рассматриваемому пути, может иметь различные знаки инверсии, знак инверсии над переменной x_i также может изменяться.

Обозначим символом D'_{rise} ДНФ, полученную из D_{rise} удалением переменной x_i , а символом D'_{fall} ДНФ, полученную из D_{fall} удалением переменной \bar{x}_i . Обозначим символом D_{rob} ДНФ, представляющую тестовые пары соседних по переменной x_i наборов для робастно тестируемых неисправностей задержек пути α : $D_{rob} = D'_{rise} \wedge D'_{fall}$. Заметим, что ДНФ D_{rob} не содержит переменной x_i .

Конъюнкции ДНФ D_{rob} таковы, что соответствующие им булевы векторы в пространстве переменных $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ задают тестовые пары в пространстве n переменных. Один из векторов пары получается приписыванием переменной x_i значения 0, а другой – значения 1. Из тестовой пары формируются тройки векторов, обнаруживающие задержки инверсных перепадов сигналов рассматриваемого пути.

Проиллюстрируем работу алгоритма на примере. Рассмотрим комбинационную схему C (рис. 1):

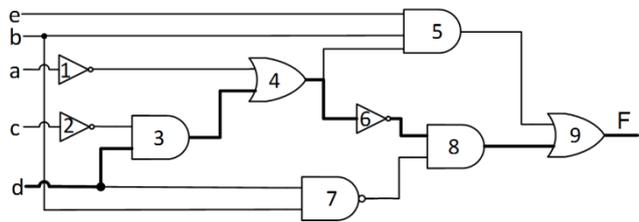


Рис. 1. Комбинационная схема C

Пусть путь α включает элементы с номерами 3, 4, 6, 8, 9. Начало пути отмечается переменной d . Таким образом, $\alpha = d, u_3, u_4, u_6, u_8, u_9$.

$$D_{u_9} / D_{u_8} = (u_5 \vee (u_8 = 0)) \oplus (u_5 \vee (u_8 = 1)) = \bar{u}_5;$$

$$D_{u_8} / D_{u_6} = ((u_6 = 0) \wedge u_7) \oplus ((u_6 = 1) \wedge u_7) = u_7;$$

$$D_{u_6} / D_{u_4} = (u_4 = 0) \oplus (u_4 = 1) = 1;$$

$$D_{u_4} / D_{u_3} = (u_1 \vee (u_3 = 0)) \oplus (u_1 \vee (u_3 = 1)) = \bar{u}_1;$$

$$D_{u_3} / d = (u_2 \wedge (d = 0)) \oplus (u_2 \wedge (d = 1)) = u_2;$$

$$D_\alpha = \bar{u}_5 \wedge u_7 \wedge 1 \wedge \bar{u}_1 \wedge u_2 = a\bar{c}\bar{b} \vee a\bar{c}\bar{d};$$

$$D_{rise} = (a\bar{c}\bar{b} \vee a\bar{c}\bar{d}) \wedge d = a\bar{c}\bar{b}d; D'_{rise} = a\bar{c}\bar{b};$$

$$D_{fall} = (a\bar{c}\bar{b} \vee a\bar{c}\bar{d}) \wedge \bar{d} = a\bar{c}\bar{d}; D'_{fall} = a\bar{c};$$

$$D_{rob} = a\bar{c}\bar{b} \wedge a\bar{c} = a\bar{c}\bar{b}.$$

Получение тестовых наборов для робастно тестируемых неисправностей с использованием ROBDD-графов представлено в работе [2].

С. Получение тестовых наборов для робастно тестируемых неисправностей с использованием КНФ логической схемы

Рассмотрим элемент u комбинационной схемы C с входами x_1, \dots, x_n . Пусть требуется найти булеву разность D_u / D_{x_i} , при этом возможны следующие ситуации:

1. Тривиальный случай: $D_u / D_{x_i} = 1$, если элемент u является одно входным (инвертор) или двухвходовым элементом вида XOR / NXOR (исключающее ИЛИ / эквивалентность):

$$D_u / D_{x_i} = (x_k \oplus (x_i = 0)) \oplus (x_k \oplus (x_i = 1)) = x_k \oplus \bar{x}_k = 1;$$

$$D_{u_i} / D_{u_j} = (x_k \sim (x_i = 0)) \oplus (x_k \sim (x_i = 1)) = \bar{x}_k \oplus x_k = 1.$$

2. $D_u / D_{x_i} = (x_1 \wedge \dots \wedge x_{i-1} \wedge x_{i+1} \wedge \dots \wedge x_n)$, если u является элементом вида AND / NAND (И / НЕ И):

$$\begin{aligned} D_u / D_{x_i} &= (x_1 \wedge \dots \wedge (x_i = 0) \wedge \dots \wedge x_n) \oplus (x_1 \wedge \dots \wedge (x_i = 1) \wedge \dots \wedge x_n) = \\ &= 0 \oplus (x_1 \wedge \dots \wedge x_{i-1} \wedge x_{i+1} \wedge \dots \wedge x_n) = (x_1 \wedge \dots \wedge x_{i-1} \wedge x_{i+1} \wedge \dots \wedge x_n); \\ D_u / D_{x_i} &= \overline{(x_1 \wedge \dots \wedge (x_i = 0) \wedge \dots \wedge x_n)} \oplus \overline{(x_1 \wedge \dots \wedge (x_i = 1) \wedge \dots \wedge x_n)} = \\ &= 1 \oplus (x_1 \wedge \dots \wedge x_{i-1} \wedge x_{i+1} \wedge \dots \wedge x_n) = (x_1 \wedge \dots \wedge x_{i-1} \wedge x_{i+1} \wedge \dots \wedge x_n). \end{aligned}$$

3. $D_u / D_{x_i} = \overline{(x_1 \vee \dots \vee x_{i-1} \vee x_{i+1} \vee \dots \vee x_n)}$, если u является элементом вида OR / NOR (ИЛИ / НЕ ИЛИ):

$$\begin{aligned} D_u / D_{x_i} &= (x_1 \vee \dots \vee (x_i = 0) \vee \dots \vee x_n) \oplus (x_1 \vee \dots \vee (x_i = 1) \vee \dots \vee x_n) = \\ &= (x_1 \vee \dots \vee x_{i-1} \vee x_{i+1} \vee \dots \vee x_n) \oplus 1 = (x_1 \vee \dots \vee x_{i-1} \vee x_{i+1} \vee \dots \vee x_n); \\ D_u / D_{x_i} &= \overline{(x_1 \vee \dots \vee (x_i = 0) \vee \dots \vee x_n)} \oplus \overline{(x_1 \vee \dots \vee (x_i = 1) \vee \dots \vee x_n)} = \\ &= \overline{(x_1 \vee \dots \vee x_{i-1} \vee x_{i+1} \vee \dots \vee x_n)} \oplus 0 = \overline{(x_1 \vee \dots \vee x_{i-1} \vee x_{i+1} \vee \dots \vee x_n)}. \end{aligned}$$

В случаях 2 и 3, если элемент u является двухвходовым (с входами x_i и x_j), выражение D_u / D_{x_i} упрощается до x_j и \bar{x}_j , соответственно.

Таким образом, выражения для булевых разностей элементов пути, и, как следствие, булеву разность D_α пути в целом можно представить в виде надстройки над исходной схемой C (рис. 2):

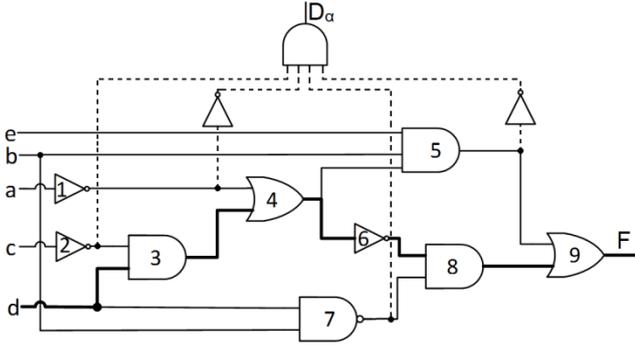


Рис. 2. Реализация булевой разности D_α в виде логической схемы C^*

КНФ элемента y комбинационной схемы C^* строится в виде т.н. КНФ Цейтина [5] по следующим правилам:

$$\begin{aligned}
 y &= \bar{x} : (x \vee y) \wedge (\bar{x} \vee \bar{y}); \\
 y &= x_1 \oplus x_2 : (\bar{x}_1 \vee x_2 \vee y) \wedge (x_1 \vee \bar{x}_2 \vee y) \wedge (x_1 \vee x_2 \vee \bar{y}) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{y}); \\
 y &= \overline{x_1 \oplus x_2} : (x_1 \vee x_2 \vee y) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee y) \wedge (\bar{x}_1 \vee x_2 \vee \bar{y}) \wedge (x_1 \vee \bar{x}_2 \vee \bar{y}); \\
 y &= \overline{x_1 \wedge x_2 \wedge \dots \wedge x_n} : (x_1 \vee \bar{y}) \wedge (x_2 \vee \bar{y}) \wedge \dots \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \dots \vee y); \\
 y &= x_1 \wedge x_2 \wedge \dots \wedge x_n : (x_1 \vee y) \wedge (x_2 \vee y) \wedge \dots \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \dots \vee \bar{y}); \\
 y &= \overline{x_1 \vee x_2 \vee \dots \vee x_n} : (\bar{x}_1 \vee y) \wedge (\bar{x}_2 \vee y) \wedge \dots \wedge (x_1 \vee x_2 \vee \dots \vee \bar{y}); \\
 y &= x_1 \vee x_2 \vee \dots \vee x_n : (\bar{x}_1 \vee \bar{y}) \wedge (\bar{x}_2 \vee \bar{y}) \wedge \dots \wedge (x_1 \vee x_2 \vee \dots \vee y).
 \end{aligned}$$

КНФ Цейтина булевой разности пути α схемы C представляет собой произведение КНФ элементов подсхемы C^* с выходом D_α (рис. 2). Полученная КНФ K_α является функцией не более чем от $(n + m)$ переменных, где n – число входов схемы C , m – число элементов подсхемы C^* с выходом D_α .

Полученная КНФ K_α содержит в себе как область единичных, так и область нулевых наборов функции D_α . Для извлечения тестовых наборов необходимо положить переменную, соответствующую выходу D_α , равной единице. Затем полученная КНФ $K_\alpha^{D_\alpha=1}$ подаётся на вход SAT-решателя. Полученное решение в виде интервала возможно дополнительно расширить. Для этого необходимо заменить одну из его компонент, сопоставляемых входам схемы C^* , неопределённым значением и добавить дизъюнкт, соответствующий данному интервалу, к области нулевых наборов K_α ($K_\alpha^{D_\alpha=0}$). Если полученная КНФ является невыполнимой, расширение интервала допустимо. При многократном обращении к SAT-решателю получаем представление D_α в виде ДНФ. Заметим, что, как правило, тестовые последовательности строятся для подмножества наиболее длинных путей заданной схемы. Как показывают эксперименты, ДНФ, представляющие D_α этих путей, можно представить ROBDD-графом. В случае, если ДНФ окажутся слишком длинными, мы можем ограничиться

подмножеством конъюнкций и соответствующими ROBDD-графами.

III. ОБЕСПЕЧЕНИЕ СНИЖЕНИЯ ПОТРЕБЛЯЕМОЙ СХЕМОЙ МОЩНОСТИ ПРИ ТЕСТИРОВАНИИ

Традиционные подходы к построению тестовых последовательностей основаны на использовании одной или нескольких тестовых пар, не обязательно соседних векторов, для каждого из рассматриваемых путей [6-11] и их перестановках с целью минимизации потребляемой мощности.

В нашем распоряжении для каждого пути имеется множество всех тестовых пар соседних булевых векторов, компактно представленных ROBDD графом R_{rob} . Важно, что ROBDD графы различных путей могут пересекаться, порождая фрагменты тестовой последовательности, сплошь состоящие из соседних векторов, то есть фрагменты, характеризующиеся минимальным потреблением мощности и минимальной длиной для соответствующего подмножества путей.

В работе [4] исследовано влияние различных подходов к пересечению ROBDD графов R_{rob} различных путей схемы. Построение тестовой последовательности для заданного множества путей сводится к поиску возможных пересечений графов. При нахождении непустых пересечений из r графов формируется фрагмент тестовой последовательности, состоящий из $(2r + 1)$ соседних булевых векторов. Фрагменты для разных подмножеств путей затем соединяются друг с другом так, чтобы при переходе от одного фрагмента к другому смежные векторы различных фрагментов были, по возможности, ближе друг к другу в смысле расстояния по Хеммингу, что приводит к снижению пиковых значений потребляемой мощности. Итак, тестовая последовательность состоит из последовательности фрагментов, каждый из которых является оптимальным в смысле потребления мощности и длины (количества тестов). Это означает, что рассмотренный подход позволяет получать более короткие тестовые последовательности для заданного множества путей и снижать как потребляемую мощность в целом, так и ее пиковые значения.

Основываясь на вышеизложенных фактах, был предложен ряд эвристических алгоритмов, целью которых является снижение потребляемой схемой мощности за счёт пересечения ROBDD-графов. Для описания предлагаемых ниже алгоритмов введём следующие обозначения:

Рассматривается многовыходная комбинационная схема C , входы которой отмечены переменными x_1, \dots, x_n , выходы – переменными y_1, \dots, y_m . Подсхемы схемы C обозначим как C_j ($j = 1, m$). Далее, путь в схеме C , начало которого соответствует переменной x_i , а выход – переменной y_j обозначим как $p_{i,j}$. ROBDD-граф, представляющий тестовые пары для тестирования пути $p_{i,j}$, обозначим через $R(p_{i,j})$; множество ROBDD-графов схемы C – $R(C)$, подмножество $R(C)$ графов путей, начало которых отмечено одной и той же переменной x_i

– $R_i(C)$. Аналогично, для подсхемы C_j соответствующие множества графов обозначим как $R(C_j)$ и $R_i(C_j)$.

Известно [3], что пересечение графов внутри множества $R_i(C_j)$ всегда пусто. Однако не ясно, в каком случае вероятность пересечения выше: при пересечении графов множеств $R_i(C_j)$ и $R_k(C_j)$ или множеств $R_i(C_j)$ и $R_i(C_i)$, а в случае смешанного подхода – порядок пересечения (пересекать сначала внутри подсхем или между подсхемами). Были разработаны следующие эвристические алгоритмы.

1) Семейство алгоритмов, пересекающих графы множеств $R_i(C_j)$ и $R_k(C_j)$. Отличительная особенность этих алгоритмов от рассмотренных ниже заключается в проведении перебора графов до тех пор, пока пересечение не будет найдено (или пока не будут перебраны все графы). Различие между алгоритмами этого семейства заключается только в ранге пересечения (т.е. количестве графов, для которых ищется одновременное пересечение.). Здесь стоит отметить, что в случае, когда пересечения заданного ранга не существует, продолжается поиск пересечения более низкого ранга.

Как следствие, данные алгоритмы дают наилучшие результаты в снижении потребляемой мощности, при этом уступая в скорости работы прочим алгоритмам.

2) Алгоритм, пересекающий в первую очередь графы разных подсхем:

$$R_1(C_1), R_1(C_2), \dots, R_1(C_m), R_2(C_1), R_2(C_2), \dots, R_2(C_m), \dots$$

В случае если на очередном шаге пересечение оказалось пустым, итоговый фрагмент тестовой последовательности строится на основе графа, полученного на предыдущем шаге.

3) Алгоритм, пересекающий в первую очередь графы внутри одной подсхемы:

$$R_1(C_1), R_2(C_1), \dots, R_n(C_1), R_1(C_2), R_2(C_2), \dots, R_n(C_2), \dots$$

В случае если на очередном шаге пересечение оказалось пустым, итоговый фрагмент тестовой последовательности строится на основе графа, полученного на предыдущем шаге.

Отметим, что построение итоговой тестовой последовательности из полученного множества ROBDD-графов для всех алгоритмов идентично:

1) Следующий тестовый интервал выбирается с помощью эвристического алгоритма, нацеленного на получение близкого в смысле расстояния по Хеммингу интервала путём прохода от корня ROBDD-графа до его 1-концевой вершины [4].

2) Полученная последовательность тестовых интервалов доопределяется до последовательности тестовых векторов таким образом, чтобы число перепадов в полученной последовательности было минимально возможным [4].

IV. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для сравнения эффективности алгоритмов были использованы схемы из набора бенчмарков ISCAS'89.

Для каждого выхода каждой схемы было выбрано не менее 10 самых длинных путей. Общая информация о бенчмарках, использованных при тестировании, представлена в табл. 1. Во втором столбце указано название схемы. В третьем, четвертом и пятом столбцах указаны количество входов (N_{in}), выходов (N_{out}) и вентилях (N_{gates}). В шестом и седьмом столбцах указано общее количество выбранных путей (N_{pt}) и количество робастно тестируемых путей (N_{pr}). В последнем столбце указано соотношение (Pr) робастно тестируемых путей к выбранным путям (в процентах).

Таблица 1

Информация об использованных бенчмарках

№ п/п	Бенчмарк	N_{in}	N_{out}	N_{gates}	N_{pt}	N_{pr}	Pr
1	s298	17	20	119	146	95	65%
2	s344	24	26	160	159	111	70%
3	s400	24	27	162	258	213	83%
4	s444	24	27	181	237	142	60%
5	s641	54	42	379	309	137	44%
6	s820	23	24	289	232	230	99%
7	s953	45	52	395	338	313	93%
8	s1196	32	32	529	334	162	49%
9	s1488	14	25	653	312	291	93%
10	s1494	14	25	647	336	306	91%

В табл. 2-5 в столбцах указаны следующие параметры: длина тестовой последовательности (SI), общее число перепадов (Pt), доля единичных перепадов (Os), пиковые перепады (Ps).

В табл. 2 представлены «контрольные» данные – пересечения ROBDD-графов не производились.

Таблица 2

Экспериментальные результаты для алгоритма без пересечений

№ п/п	SI	Pt	Os	Ps
1	273	331	70%	6
2	312	360	74%	3
3	612	755	69%	6
4	401	471	74%	5
5	350	358	91%	3
6	597	779	64%	6
7	896	1309	51%	7
8	467	652	56%	6
9	823	1146	56%	6
10	877	1242	55%	8

В табл. 3 представлены сравнительные данные семейства алгоритмов, в основе которого лежит пересечение ROBDD-графов рангов 2-5. Отметим, что во всех случаях непересечённых графов не осталось.

В табл. 4 и табл. 5 помимо аналогичных данных (для алгоритмов 2 и 3 из раздела 4) приведены количества графов, которые удалось (Ni) и не удалось (Nni) пересечь, а также наибольший ранг пересечения (Rm).

Таблица 3

Экспериментальные результаты для алгоритмов с пересечениями ранга 2-5

№	Sl	Pt	Os	Ps
1	252	290	77%	3
2	289	326	78%	3
3	568	679	75%	7
4	364	427	76%	5
5	344	362	85%	3
6	579	733	70%	7
7	815	1110	61%	9
8	442	589	63%	7
9	798	1057	64%	7
10	843	1141	62%	6

Таблица 4

Экспериментальные результаты для алгоритма с пересечениями графов для путей с разными выходами

№ п/п	Sl	Pt	Os	Ps	Ni	Nni	Rm
1	236	307	66%	5	26	15	6
2	269	352	67%	10	27	15	7
3	571	766	62%	7	56	80	6
4	364	482	62%	8	27	52	8
5	313	339	87%	5	24	14	30
6	667	978	52%	6	31	165	4
7	907	1467	45%	8	32	243	3
8	432	703	48%	8	48	54	4
9	845	1340	46%	6	37	215	3
10	887	1383	48%	7	41	221	3

Таблица 5

Экспериментальные результаты для алгоритма с пересечениями графов для путей с разными входами

№ п/п	Sl	Pt	Os	Ps	Ni	Nni	Rm
1	241	277	79%	5	33	7	6
2	274	335	71%	4	34	14	9
3	539	642	75%	5	76	15	6
4	353	415	75%	5	55	5	7
5	328	357	84%	4	38	6	11
6	579	715	70%	6	53	46	9
7	778	1032	64%	7	82	49	9
8	436	605	60%	7	50	36	6
9	804	1088	60%	6	75	107	7
10	846	1140	61%	7	82	109	6

V. ЗАКЛЮЧЕНИЕ

Из экспериментальных данных можно сделать вывод о том, что последний предложенный алгоритм (пересекающий в первую очередь графы внутри одной подсхемы) имеет наилучшие показатели. С одной стороны, он обеспечивает лучшие показатели в сокращении длины (~10%) и числа перепадов (10-15%). Это является следствием высокого процента пересечений среди ROBDD-графов (около 80-85%). При этом вычислительные затраты для выполнения алгоритма крайне низкие ($O(n)$, где n – число путей). Алгоритм поиска определённого ранга пересечения (табл. 3) при повышении ранга пересечения способен дать лучшие результаты, однако будет сильно проигрывать по скорости выполнения ввиду возможного перебора графов.

Проведены предварительные эксперименты над схемами, для которых D_a строится в виде ДНФ с помощью SAT-решателя, поскольку построение и перемножение ROBDD-графов, необходимые для нахождения D_a , не удастся выполнить из-за размеров графов. В том случае, когда использование ROBDD-графов затруднительно ввиду их больших размеров, существует возможность построить КНФ Цейтина [5] с последующим нахождением одной, нескольких или всех тестовых пар для выбранного пути через SAT решатель. Таким образом, при тестировании схем появляется возможность выбирать подходящее соотношение энергопотребления тестовой последовательности к времени, затраченному на её получение.

В целом, предлагаемый в работе подход, помимо обнаружения всех робастно тестируемых неисправностей задержек путей, позволяет существенно снизить потребляемую схемой мощность при её тестировании.

ЛИТЕРАТУРА

- [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] A.Yu.Matrosova, V.V. Andreeva, E.A. Nikolaeva. Finding Test Pairs for PDFs in Logic Circuits Based on Using Operations on ROBDDs //Russian Physics Journal. 2018. Vol. 61, № 5. pp. 994-999.
- [3] A.Yu.Matrosova, V.V. Andreeva, E.A., Tychinskiy V.Z., Goshin G.G. Applying ROBDDs for delay testing of logical circuits. Izvestia vyzov. Physics. 2019.v. 62, № 5.pp. 86-94.
- [4] Matrosova A., Andreeva V., Tychinskiy V. Deriving Low Power Test Sequences Detecting Robust Testable PDFs //Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS), 13-16 september 2019, Batumi. Kharkov: IEEE, 2019. P. 406-409.
- [5] Цейтин Г. С. О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. – 1968. – Т. 8. – С. 234-259.
- [6] P. Lindgren, M. Kerttu, M. Thornton and R. Drechsler “Low power optimization technique for BDD mapped circuits” ASP-DAC 2001, pp. 615-621.

- [7] R.S. Shelar, S.S. Sapatnekar "An efficient algorithm for low power pass transistor logic synthesis" ASP-DAC 2002, pp. 87-92.
- [8] G. Gekas, D. Nikolos, E. Kalligeros, X. Kavousianos, "Power aware test-data compression for scan-based testing", ICECS 2005. 12th IEEE International Conference on, pp. 1-4.
- [9] J.T. Tudu, E. Larsson, V. Singh, V.D. Agrawal, "On Minimization of Peak Power for Scan Circuit during Test", Test Symposium 2009 14th IEEE European, 2009, pp. 25-30.
- [10] Z. Kotasek, J. Skarvada, J. Strnadel, "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, pp. 364-369.
- [11] V. Sinduja, S. Raghav, J. P Anita, "Efficient don't-care filling method to achieve reduction in test power", ICACCI, 2015, pp. 478-482.

Deriving Low Power Robust PDFs Based on Applying SAT-Solvers and Operations on ROBDDs

A.Yu. Matrosova, V.Z. Tychinskiy, V.V. Andreeva

Institute of applied mathematics and computer science, National Research Tomsk State University,
Tomsk, mau11@yandex.ru, tvz.041@gmail.com, avv.21@mail.ru

Abstract — new approach to deriving low power test sequences that detect robust testable PDFs in logical circuits is suggested. Decreasing power consumption is provided by decreasing the number of switches during testing and by reducing length of the test sequence. The approach is based on finding all test pairs consisting of neighbor Boolean vectors for a circuit path. The test pairs are compactly represented by the proper ROBDD [2]. Each pair of neighbor Boolean vectors generates three neighbor Boolean vectors that detect robust PDFs of both rising and falling transitions of a path. Current approaches are oriented to finding if only one test pair both for rising and falling transition of a circuit path there exists. Applying all test pairs of neighbor Boolean vectors and using properties of ROBDDs representing these pairs we provide addition facility to cut power consumption of the test sequences. The general principles of deriving the test sequences are based on intersection of ROBDDs of different paths [4] and a procedure [4] oriented to cutting Hemming distance between neighbor vectors of different ROBDDs. Some experimental results for the longest circuit paths are represented. They demonstrate the high quality of test sequences in terms of power consumption, not achievable by traditional methods. In the case where the use of ROBDDs is difficult due to their large sizes, an alternative approach is proposed, which is based on the obtaining of the Tseitin CNF [5]. The complexity of this CNF deriving linearly depends on the number of gates in the circuit. The obtained CNF is fed to the input of the SAT solver, which allows getting one, several or all test pairs for the selected path. Thus, when testing circuits, it becomes possible to find the proper ratio between the power consumption of the test sequence and the time taken to obtain it.

Keywords — robust testable Path-Delay Faults (PDFs), reduced ordered binary decision diagrams (ROBDD), SAT-solvers, combinational circuits, sequential circuits.

REFERENCES

- [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] A.Yu.Matrosova, V.V. Andreeva, E.A. Nikolaeva. Finding Test Pairs for PDFs in Logic Circuits Based on Using Operations on ROBDDs //Russian Physics Journal. 2018. Vol. 61, № 5. pp. 994-999.
- [3] A.Yu.Matrosova, V.V. Andreeva, E.A., Tychinskiy V.Z., Goshin G.G. Applying ROBDDs for delay testing of logical circuits. Izvestia vyzov. Physics. 2019.v. 62, № 5.pp. 86-94.
- [4] Matrosova A., Andreeva V., Tychinskiy V. Deriving Low Power Test Sequences Detecting Robust Testable PDFs //Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS), 13-16 september 2019, Batumi. Kharkov: IEEE, 2019. P. 406-409.
- [5] Tseitin G. S. O slozhnosti vyvoda v ischislenii vyskazyvanij (On the Complexity of Derivation in Propositional Calculus) // Zapiski nauchnyh seminarov LOMI AN SSSR. – 1968. – T. 8. – C. 234-259.
- [6] P. Lindgren, M. Kerttu, M. Thornton and R. Drechsler "Low power optimization technique for BDD mapped circuits" ASP-DAC 2001, pp. 615-621.
- [7] R.S. Shelar, S.S. Sapatnekar "An efficient algorithm for low power pass transistor logic synthesis" ASP-DAC 2002, pp. 87-92.
- [8] G. Gekas, D. Nikolos, E. Kalligeros, X. Kavousianos, "Power aware test-data compression for scan-based testing", ICECS 2005. 12th IEEE International Conference on, pp. 1-4.
- [9] J.T. Tudu, E. Larsson, V. Singh, V.D. Agrawal, "On Minimization of Peak Power for Scan Circuit during Test", Test Symposium 2009 14th IEEE European, 2009, pp. 25-30.
- [10] Z. Kotasek, J. Skarvada, J. Strnadel, "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, pp. 364-369.
- [11] V. Sinduja, S. Raghav, J. P Anita, "Efficient don't-care filling method to achieve reduction in test power", ICACCI, 2015, pp. 478-482.