

ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 004.312

А.Ю. Матросова, С.А. Останин, А.В. Бухаров, И.Е. Кириенко

ПОИСК ВСЕХ ТЕСТОВЫХ НАБОРОВ ДЛЯ НЕИСПРАВНОСТИ ЛОГИЧЕСКОЙ СХЕМЫ И ПРЕДСТАВЛЕНИЕ ИХ ROBDD-ГРАФОМ

Работа выполнена при финансовой поддержке РНФ в рамках научного проекта (№ 14-19-00218).

Рассматриваются комбинационная схема и константная неисправность на полюсе одного из ее вентилях. Предлагается алгоритм построения множества всех тестовых наборов для заданной неисправности, ориентированный на компактное представление этого множества в виде ROBDD-графа. Алгоритм основан на методе достраивания конъюнкций, предназначенном для построения одного (любого) тестового набора для заданной неисправности комбинационной схемы. Его расширение на построение всех тестовых наборов, в том числе и комбинационных эквивалентов схем с памятью, обеспечивается применением операций над ROBDD-графами.

Ключевые слова: одиночные константные неисправности; тестовые наборы; ROBDD-графы.

Построение проверяющих тестов для комбинационных схем в классе одиночных константных неисправностей сводится к нахождению одного тестового набора для неисправности рассматриваемого класса. Последняя проблема широко исследовалась в России и за рубежом, были разработаны различные методы ее решения [1]. Нахождение всех тестовых наборов для неисправности из заданного класса может потребоваться, например, при поиске проверяющего теста минимальной длины. Решение обеих задач традиционными методами, т.е. поиск всех тестовых наборов и построение проверяющего теста минимальной длины, связано с большими вычислительными затратами и потребностью хранить огромные объемы информации. Поэтому на практике ограничиваются поисками нескольких или одного тестового набора для заданной неисправности и построением минимизированных проверяющих тестов, достаточно близких к минимальным.

В приложениях, связанных с тестированием неисправностей задержек путей схемы [2, 3], требуется находить все тестовые наборы, порождаемые конкретной конъюнкцией ДНФ, описывающей поведение схемы.

Известно, что поведение подсхемы в условиях включения ее в схему может быть описано частичной булевой функцией [4, 5]. Этот факт используется для минимизации схемы в целом. Множества единичных и нулевых наборов частичной функции подсхемы могут, на наш взгляд, быть представлены с помощью всех тестовых наборов для соответствующих константных неисправностей.

Один из подходов к синтезу частично программируемых логических схем высокой производительности [6] основан на описании поведения подсхемы, включенной в заданную схему, в виде реализации частичной функции. Реализаций может быть множество. Точное знание частичной функции позволяет найти более качественную в том или ином смысле реализацию и определить места включения в схему из вентилях программируемых блоков с целью маскирования возможных неисправностей. Речь идет о неисправностях, обнаруживаемых на заключительных этапах проектирования схем. Для коррекции схемы необходимо вернуться на ранние этапы проектирования, что существенно увеличивает стоимость проектирования схемы. Присутствующие в схеме резервные программируемые блоки позволяют маскировать обнаруженную неисправность и тем самым избежать возвращения к

ранним этапам проектирования. Исходя из вышесказанного, важно уметь находить множество всех тестовых наборов для заданной неисправности и компактно его представлять.

В данной работе предлагается находить множество всех тестовых наборов для константной неисправности на полюсе логического элемента на основе разработанного нами ранее метода достраивания конъюнкций. Используется представление схем и подсхем в виде ROBDD-графов. Построение множества всех тестовых наборов выполняется путем перемножения ROBDD-графов подсхем заданной схемы. Как известно, графовое представление булевых функций является более компактным по сравнению с аналитическими способами задания функций в виде ДНФ или КНФ, а операции над ROBDD-графами обладают полиномиальной сложностью.

В 1-м разделе работы обсуждается метод достраивания конъюнкций, что необходимо для понимания предлагаемого нами алгоритма построения всех тестовых наборов для заданной неисправности в комбинационной схеме. Во 2-м разделе предлагается алгоритм построения множества всех тестовых наборов, приводится пример, его иллюстрирующий, и обсуждается распространение алгоритма на комбинационный эквивалент схемы с памятью в условиях описания поведения последовательностной схемы системой частичных булевых функций.

1. Метод достраивания конъюнкций для построения одного (любого) тестового набора

В работе [1] нами был предложен метод достраивания конъюнкций, в котором выявлены возможности сокращения перебора при поиске одного (любого) тестового набора для одиночной константной неисправности полюса логического элемента схемы. Предполагается, хотя это и не обязательно, что схема построена из вентилях OR, NOR, AND, NAND, NOR, EXOR.

Заданы одна выходная комбинационная схема и неисправный полюс v . Обозначим через $\varphi(X)$, $\varphi^v(X)$ булевы функции, реализуемые схемой в исправном и неисправном состояниях, где $X = \{x_1, \dots, x_n\}$ – множество входных переменных схемы. Отыскание теста для данной неисправности сведем к отысканию корня булева уравнения

$$D(\varphi)\bar{D}(\varphi^v) \vee \bar{D}(\varphi)D(\varphi^v) = 1.$$

По заданной схеме C (рис. 1) и заданному неисправному полюсу v выделим подсхему (рис. 2), реализующую функцию $\zeta(X, v)$, при условии, что переменная v рассматривается в качестве входной наряду с переменными x_1, \dots, x_n . Выделим также подсхему (рис. 3), выходом которой является неисправный полюс v , а входами – переменные x_1, \dots, x_n . Эта подсхема реализует функцию $f(X)$.

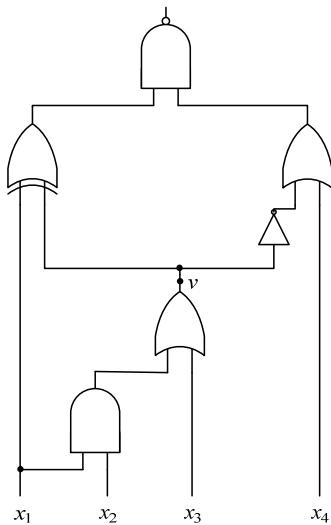


Рис. 1. Комбинационная схема C

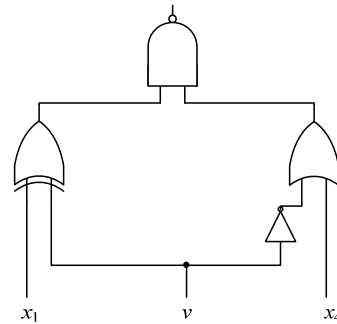


Рис. 2. Подсхема, реализующая функцию $\zeta(X, v)$

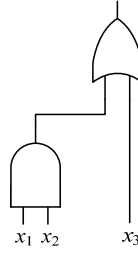


Рис. 3. Подсхема, реализующая функцию $f(X)$

Для схемы на рис. 2 функция $\zeta(X, \nu)$ в виде ДНФ представляется выражением $x_1\nu \vee \bar{x}_1\bar{\nu} \vee \bar{x}_4\nu$, а функция $f(X)$ – выражением $x_1x_2 \vee x_3$.

Обозначим через σ тип неисправности на полюсе ν , $\sigma \in \{0, 1\}$.

Из построения функций $\zeta(X, \nu)$ и $f(X)$ следует, что

$$\begin{aligned}\varphi(X) &= \zeta(X, f(X)), \\ \varphi^\nu(X) &= \zeta(X, \sigma).\end{aligned}$$

Обозначим через $D(\zeta)$ ДНФ функции $\zeta(X, \nu)$. Представим эту ДНФ в виде $K \vee K^\nu \vee K^{\bar{\nu}}$; в ДНФ K – конъюнкции, не содержащие литер $\nu, \bar{\nu}$; в ДНФ K^ν включаются конъюнкции, содержащие литеру ν ; в ДНФ $K^{\bar{\nu}}$ – конъюнкции, содержащие литеру $\bar{\nu}$. Если $\sigma = 0$ (1), то K^ν ($K^{\bar{\nu}}$) обращается в ноль, а $K^{\bar{\nu}}$ (K^ν) обращается в $K^{\bar{\nu}'}$ ($K^{\nu'}$), причем ДНФ $K^{\bar{\nu}'}$ ($K^{\nu'}$) получается из ДНФ $K^{\bar{\nu}}$ (K^ν) вычеркиванием литеры $\bar{\nu}$ (ν). Тогда для $\sigma = 0(1)$ $D(\varphi^\nu) = K \vee K^{\bar{\nu}'}$ ($K \vee K^{\nu'}$).

Пусть ДНФ $K^{\bar{\nu}''}$ ($K^{\nu''}$) получается из $K^{\bar{\nu}}$ (K^ν) подстановкой вместо литеры $\bar{\nu}$ (ν) функции $\bar{f}(X)$ ($f(X)$), представленной в виде ДНФ. Обозначим через β набор значений переменных x_1, \dots, x_n .

Теорема 1. β – тест для неисправности $\sigma = 0(1)$ и $\varphi(\beta) = 1$, если и только если

$$\begin{aligned}K^{\nu''}(\beta) &= 1 \quad (K^{\bar{\nu}''}(\beta) = 1), \\ K(\beta) &= 0, \\ K^{\bar{\nu}'}(\beta) &= 0 \quad (K^{\nu'}(\beta) = 0).\end{aligned}$$

Доказательство этой теоремы приведено в работе [1].

Пусть $\bar{\varphi}(X)$, $\bar{\varphi}^\nu(X)$, $\bar{\zeta}(X, \nu)$ – инверсии функций $\varphi(X)$, $\varphi^\nu(X)$, $\zeta(X, \nu)$. Обозначим через $\bar{D}(\zeta)$ ДНФ функции $\bar{\zeta}(X, \nu)$.

Представим эту ДНФ в виде $\underline{K} \vee \underline{K}^\nu \vee \underline{K}^{\bar{\nu}}$.

В ДНФ \underline{K} включаются конъюнкции, не содержащие литер $\nu, \bar{\nu}$; в ДНФ \underline{K}^ν – конъюнкции, содержащие литеру ν ; в ДНФ $\underline{K}^{\bar{\nu}}$ включаются конъюнкции, содержащие литеру $\bar{\nu}$. Если $\sigma = 0(1)$, то \underline{K}^ν ($\underline{K}^{\bar{\nu}}$) обращается в ноль, а $\underline{K}^{\bar{\nu}}$ (\underline{K}^ν) обращается в $\underline{K}^{\bar{\nu}'}$ ($\underline{K}^{\nu'}$), причем $\underline{K}^{\bar{\nu}'}$ ($\underline{K}^{\nu'}$) получается из $\underline{K}^{\bar{\nu}}$ (\underline{K}^ν) вычеркиванием литеры $\bar{\nu}$ (ν). Тогда для $\sigma = 0(1)$ $\bar{D}(\varphi^\nu) = \underline{K} \vee \underline{K}^{\bar{\nu}'}$ ($\underline{K} \vee \underline{K}^{\nu'}$).

Пусть ДНФ $\underline{K}^{\bar{\nu}''}$ ($\underline{K}^{\nu''}$) получается из $\underline{K}^{\bar{\nu}}$ (\underline{K}^ν) подстановкой вместо литеры $\bar{\nu}$ (ν) функции $\bar{f}(X)$ ($f(X)$), представленной в виде ДНФ. Тогда $\bar{D}(\varphi) = \underline{K} \vee \underline{K}^{\bar{\nu}''} \vee \underline{K}^{\nu''}$.

Теорема 2. β – тест для неисправности $\sigma = 0(1)$ и $\varphi(\beta) = 0$, если и только если

$$\begin{aligned}\underline{K}^{\nu''}(\beta) &= 1 \quad (\underline{K}^{\bar{\nu}''}(\beta) = 1), \\ \underline{K}(\beta) &= 0, \\ \underline{K}^{\bar{\nu}'}(\beta) &= 0 \quad (\underline{K}^{\nu'}(\beta) = 0).\end{aligned}$$

Доказательство аналогично доказательству теоремы 1.

Алгоритм достраивания конъюнкций для $\sigma = 0(1)$

1. Строим ДНФ $D(\zeta)$.
2. Получаем ДНФ $K^{v''} (K^{\bar{v}''})$.
3. Выбираем очередную конъюнкцию k из $K^{v''} (K^{\bar{v}''})$. Приписываем к ней буквы таким образом, чтобы полученная конъюнкция k^* была ортогональна всем конъюнкциям из $K \vee K^{\bar{v}'} (K \vee K^{v'})$. Если k^* построить не удастся, возвращаемся к п. 3 алгоритма, с тем чтобы выбрать новую конъюнкцию из $K^{v''} (K^{\bar{v}''})$. Построив k^* , переходим к п. 5 алгоритма. Если все конъюнкции из $K^{v''} (K^{\bar{v}''})$ просмотрены, переходим к п. 4 алгоритма.
4. Возвращаемся к п. 3 алгоритма, рассматривая вместо ДНФ $D(\zeta)$ ДНФ $\bar{D}(\zeta)$, а вместо ДНФ K , $K^{\bar{v}''}$, $K^{v''}$, $K^{v'}$, $K^{\bar{v}'}$, соответственно ДНФ \underline{K} , $\underline{K}^{\bar{v}''}$, $\underline{K}^{v''}$, $\underline{K}^{v'}$, $\underline{K}^{\bar{v}'}$. Если все конъюнкции из $\underline{K}^{v''} (\underline{K}^{\bar{v}''})$ просмотрены, а тест не найден, значит, теста для рассматриваемой неисправности не существует.
5. Достраиваем конъюнкцию k^* произвольным образом до конъюнкции ранга n . Полученная конъюнкция представляет тестовый набор β .

Достраивание конъюнкции k до k^* сводится к отысканию одного корня булева уравнения $D = 0$. Левая часть уравнения получается из не ортогональных для k конъюнкций, принадлежащих ДНФ $K \vee K^{\bar{v}'} (K \vee K^{v'})$ или ДНФ $\underline{K} \vee \underline{K}^{\bar{v}'} (\underline{K} \vee \underline{K}^{v'})$, причем в этих конъюнкциях вычеркнуты литеры, используемые в k . При отыскании тестового набора описываемым здесь методом приходится достраивать очередную конъюнкцию k до k^* , пока k^* не будет найдена.

Рассмотрим пример.

$$D(\zeta) = x_1 v \vee \bar{x}_1 \bar{v} \vee \bar{x}_4 v, \sigma = 0, K^v = x_1 v \vee \bar{x}_4 v, K^{\bar{v}} = \bar{x}_1 \bar{v}, K^{\bar{v}'} = \bar{x}_1.$$

$$D(f) = x_1 x_2 \vee x_3, K^{v''} = x_1 x_2 \vee x_1 x_3 \vee x_3 \bar{x}_4, k = k^* = x_1 x_2, \beta = 1 \ 100.$$

Метод достраивания конъюнкций требует получения ДНФ функций $D(\zeta)$, $D(f)$. Пользуясь этими ДНФ и их инверсиями, можно найти тестовый набор для данной неисправности, не зная ДНФ $D(\varphi)$ исправной функции.

2. Метод достраивания конъюнкций для построения всех тестовых наборов

Воспользуемся идеями, которые лежат в основе алгоритма построения тестового набора для заданной неисправности, с тем чтобы найти все ее тестовые наборы и компактно их представить.

Сначала опишем процедуру выделения из исходной одной выходной схемы S подсхемы, реализующей функцию $\zeta(X, v)$, для которой заданный неисправный полюс является входным. Для этого выбрасываем из схемы элемент e , выход которого является полюсом v , и все связанные только с ним элементы схемы, а именно элементы, выходы которых связаны только с входами элемента e . Они образуют 1-й ярус. Пусть уже построено i ярусов. Тогда в $(i+1)$ -й ярус включаются элементы, такие что выходы каждого элемента соединены только с входами элементов i -го и / или любых из предшествующих ярусов. Найденные таким образом элементы убираем из исходной схемы и получаем требуемую подсхему с входным полюсом v . Если неисправный полюс является входным для элемента e схемы, тогда рассматриваем связанный с ним выход предшествующего элемента g . Если элемент g не является точкой ветвления, то удаляем этот элемент вместе со связанными только с ним элементами схемы описанным выше способом. Если элемент g является точкой ветвления, то обрываем линию связи его с элементом e . Вход элемента e объявляем входом v схемы, реализующей функцию $\zeta(X, v)$.

Как известно, BDD (Binary Decision Diagram) – это ориентированный граф без циклов. Его построение основано на разложении Шеннона, используемого в каждой текущей вершине v строящегося BDD-графа:

$$f_v = \bar{x}_i f_v^{x_i=0} \vee x_i f_v^{x_i=1}.$$

Здесь f_v – функция, реализуемая в полюсе v . Пунктирная дуга (0-дуга), исходящая из v , заходит в дочернюю вершину, в которой реализуется функция $f_v^{x_i=0}$. Сплошная дуга (1-дуга) заходит в вершину, в которой реализуется функция $f_v^{x_i=1}$. BDD называется *упорядоченным* графом (Ordered BDD), если порядок разложения по переменным фиксирован для функции в целом и не зависит от конкретной вершины, к которой применяется разложение. BDD называется *сокращенным* графом (Reduced BDD), если он не содержит ни изоморфных подграфов, ни таких внутренних полюсов v , для которых выполняется условие $f_v^{x_i=0} = f_v^{x_i=1}$. Сокращенный и упорядоченный BDD (ROBDD) является каноническим представлением булевой функции относительно выбранного порядка разложения по переменным. Любой путь, связывающий корень BDD с 1-концевой вершиной, представляет конъюнкцию, а все такие пути – ортогональную ДНФ (ОДНФ) некоторой функции δ , представляемой ROBDD. ОДНФ есть дизъюнкция попарно ортогональных конъюнкций. Конъюнкции ортогональны, если в одной из них некоторая переменная присутствует со знаком инверсии, а в другой – без инверсии.

Обозначим через R^c ROBDD-граф, представляющий функцию $\zeta(X,v)$. Будем иметь в виду, что конъюнкции, сопоставляемые путям, идущим от корня этого графа через 1-дугу корня и заканчивающиеся в 1-концевой вершине, представляют ОДНФ K^v . Конъюнкции, сопоставляемые путям, идущим от корня графа R^c через 0-дугу корня и заканчивающиеся в 1-концевой вершине, представляют ОДНФ $K^{\bar{v}}$. ОДНФ, соответствующая R^c , представляется в виде $\zeta(X,v) = K^v \vee K^{\bar{v}}$, т.е. в ОДНФ отсутствуют конъюнкции, не содержащие литер v, \bar{v} . ОДНФ $K^{v'}$ представляется отдельным ROBDD-графом $R^{v'}$, корнем которого является вершина, в которую заходит 1-дуга графа R^c . ОДНФ $K^{\bar{v}'}$ представляется отдельным ROBDD-графом $R^{\bar{v}'}$, корнем которого является вершина, в которую заходит 0-дуга графа R^c .

Обозначим через R_f ROBDD-граф, представляющий функцию f . Будем иметь в виду, что конъюнкции, сопоставляемые путям, идущим от корня этого графа и заканчивающиеся в 1-концевой вершине, представляют ОДНФ функции f . Конъюнкции, сопоставляемые путям, идущим от корня этого графа и заканчивающиеся в 0-концевой вершине, представляют ОДНФ функции \bar{f} .

Сначала предлагается вместо достраивания конъюнкции k из $K^{v'}$ ($\sigma = 0$) до k^* , как это делается в приведенном выше алгоритме, достраивать конъюнкцию k из $K^{v'}$ до всевозможных k' , обеспечивая для каждой k' ортогональность всем конъюнкциям ДНФ $K^{\bar{v}'}$. Достраивание выполняется для каждой конъюнкции из $K^{v'}$. Обозначим через $\underline{R}^{\bar{v}'}$ граф, представляющий инверсию ДНФ $K^{\bar{v}'}$. Он получается из $R^{\bar{v}'}$ переименованием концевых вершин. Результат достраивания представляется графом R' , получаемым перемножением графов $R^{v'}$ и $\underline{R}^{\bar{v}'}$, т.е. $R' = R^{v'} \underline{R}^{\bar{v}'}$.

Получение всевозможных k^* , задающих тестовые наборы для неисправности $\sigma = 0$ при условии $\varphi(X) = 1$, далее обеспечивается перемножением всех полученных из $K^{v'}$ конъюнкций k' , представленных ROBDD R' , и графа R_f , задающего функцию f : $R^{t_1} = R_f R'$.

Алгоритм достраивания конъюнкций для построения всех тестовых наборов

1. Строим подсхему, реализующую функцию $\zeta(X,v)$, в которой полюс v является ее входом.
2. Строим ROBDD R^c для этой подсхемы, выбирая полюс v в качестве первого элемента в упорядоченном списке разложения по переменным. Представленный граф задает функцию $\zeta(X,v)$ и ее инверсию.
3. Строим ROBDD-графы $R_f, R^{v'}, R^{\bar{v}'}$.
4. Рассматриваем неисправность на полюсе v : $\sigma = 0$.
5. Получаем граф $R^{t_1} = R^{v'} \underline{R}^{\bar{v}'} R_f$ (конъюнкции k из $K^{v'}$ достраиваются до конъюнкций k^*), представляющий своими путями от корня к 1-концевой вершине все тестовые наборы, на которых исправная схема принимает значение 1, а неисправная – 0 в присутствии неисправности $\sigma = 0$ на полюсе v .

6. Меняем местами концевые вершины графа R^c . Получаем граф \underline{R}^c . Выполняем п. 5, рассматривая вместо ДНФ $K^{v'}$, $K^{\bar{v}}$ ДНФ $\underline{K}^{v'}$, $\underline{K}^{\bar{v}}$. Поменяв местами концевые вершины графа $R^{v'}$, получаем граф $\underline{R}^{v'}$, а затем ROBDD $R^{t_0} = \underline{R}^{v'} \underline{R}^{\bar{v}'} R_f$, представляющий своими путями от корня к 1-концевой вершине все тестовые наборы, на которых исправная схема принимает значение 0, а неисправная – 1 в присутствии неисправности $\sigma = 0$ на полюсе v .

7. Объединяем R^{t_0} и R^{t_1} графы, получаем граф $\mathbf{R}^0 = R^{t_0} \vee R^{t_1}$, представляющий своими путями от корня к 1-концевой вершине все тестовые наборы для неисправности $\sigma = 0$ на полюсе v .

8. Рассматриваем неисправность $\sigma = 1$ на полюсе v .

9. Строим ROBDD-графы \underline{R}_f (меняя местами концевые вершины графа R_f), $R^{v'}$, $R^{\bar{v}'}$.

10. Получаем граф $R^{t_1} = R^{\bar{v}'} \underline{R}^{v'} \underline{R}_f$ (конъюнкции k из $K^{\bar{v}'}$ достраиваются до конъюнкций k^*), представляющий своими путями от корня к 1-концевой вершине все тестовые наборы, на которых исправная схема принимает значение 1, а неисправная – 0 в присутствии неисправности $\sigma = 1$ на полюсе v .

11. Меняем местами концевые вершины графа R^c . Получаем граф \underline{R}^c . Выполняем п. 10, рассматривая вместо ДНФ $K^{v'}$, $K^{\bar{v}}$ ДНФ $\underline{K}^{v'}$, $\underline{K}^{\bar{v}}$. Поменяв местами концевые вершины графа $R^{\bar{v}'}$, получаем граф $\underline{R}^{\bar{v}'}$, а затем ROBDD $R^{t_0} = \underline{R}^{\bar{v}'} R^{v'} \underline{R}_f$, представляющий своими путями от корня к 1-концевой вершине все тестовые наборы, на которых исправная схема принимает значение 0, а неисправная – 1 в присутствии неисправности $\sigma = 1$ на полюсе v .

12. Объединяем графы R^{t_0} , R^{t_1} , получаем граф $\mathbf{R}^1 = R^{t_0} \vee R^{t_1}$, представляющий своими путями от корня к 1-концевой вершине все тестовые наборы для неисправности $\sigma = 1$ на полюсе v .

Рассмотрим пример. ДНФ $D(\zeta)$ представлена выражением $x_1 v \vee \bar{x}_1 \bar{v} \vee \bar{x}_4 v$.

Соответствующий ей граф (рис. 4) построен с использованием следующего порядка разложения по переменным: v, x_1, x_2, x_3, x_4 . ROBDD R_f (рис. 5) и все последующие графы строятся для того же порядка разложения по переменным:

$$D(f) = x_1 x_2 \vee x_3.$$

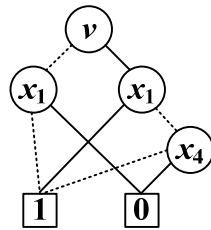


Рис. 4. ROBDD R^c для $D(\zeta)$

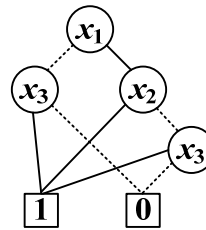


Рис. 5. ROBDD R_f для $D(f)$

Пусть $\sigma = 0$. Выделяем графы $R^{v'}$, $R^{\bar{v}'}$ из R^c . Граф $\underline{R}^{\bar{v}'}$ получается из $R^{v'}$ переименованием вершин. Граф $R^{t_1} = R^{v'} \underline{R}^{\bar{v}'} R_f$ представлен в виде ROBDD на рис. 6.

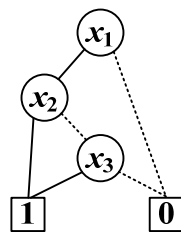


Рис. 6. ROBDD R^{t_1}

Меняем местами концевые вершины графа R^c , получаем граф \underline{R}^c .

Граф $R^{t0} = \underline{R}^{v'} R^{\bar{v}'} R_f$ представлен на рис. 7. ROBDD \mathbf{R}^0 представлен рис. 8.

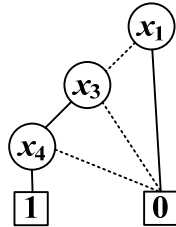


Рис. 7. ROBDD R^{t0}

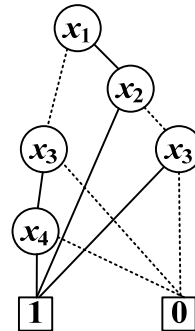


Рис. 8. ROBDD \mathbf{R}^0

Предложенный метод может быть распространен на многовыходные комбинационные схемы. Если рассматриваемый неисправный полюс соединен с несколькими выходами схемы, то для каждого из выходов строится свой граф \mathbf{R}^0 (\mathbf{R}^1), а затем находится ROBDD-граф, представляющий их объединение.

Построение множества всех тестовых наборов в схемах с памятью

Пусть комбинационная схема является комбинационной составляющей схемы с памятью (последовательностной схемы). Задание на синтез такой схемы, как правило, представляется системой частичных булевых функций. Каждая из функций φ задается множеством единичных и множеством нулевых наборов $M^1(\varphi)$, $M^0(\varphi)$. Эти наборы задают рабочую область функционирования последовательностной схемы. Поведение схемы вне рабочей области интереса не представляет. Поэтому множество тестовых наборов для заданной неисправности комбинационной составляющей последовательностной схемы следует ограничить рабочей областью функционирования схемы с памятью.

Зададим множества $M^1(\varphi)$, $M^0(\varphi)$ ROBDD-графами R_φ^1 , R_φ^0 соответственно. Тогда R^{t1} , R^{t0} представляются в виде $R^{t1} = R_\varphi^1 R^{v'} \underline{R}^{\bar{v}'} R_f$, $R^{t0} = R_\varphi^0 \underline{R}^{v'} R^{\bar{v}'} R_f$ для неисправности $\sigma = 0$. Аналогичным образом находятся множества тестовых наборов для неисправности $\sigma = 1$.

Заключение

Предложен алгоритм нахождения всех тестовых наборов для константной неисправности полюса логического элемента комбинационной схемы или комбинационного эквивалента последовательностной схемы. Алгоритм позволяет представлять множество всех тестовых наборов для неисправного полюса в виде ROBDD-графа и сводится к перемножению ROBDD-графов, соответствующих подсхемам заданной схемы.

ЛИТЕРАТУРА

1. Матросова А.Ю. Метод обнаружения неисправности в синхронном устройстве // Автоматика и телемеханика. 1977. № 12. С. 129–137.
2. Matrosova A.Yu., Melnikov A.V., Mukhamedov R.V., Ostanin S.A., Singh V. Selection of the flip-flops for partial enhanced scan techniques // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2012. № 2(19). С. 112–120.
3. Matrosova A.Yu., Ostanin S.A., Melnikov A.V., Singh V. Observability Calculation of State Variable Oriented to Robust PDFs and LOC or LOS Techniques // Proc. of IEEE East-West Design&Test Symposium (EWDTS 2012). Ukraїn, 2012. P. 155–160.

4. Muroga S., Kambayashi Y., Lai H.C., Culliney J.N. The transduction method-design of logic networks based on permissible functions // IEEE Trans. on Computers. 1989. V. 38, No. 10. P. 1404–1424.
5. Yamashita S., Sawada H., Nagoya A. SPFD: A new method to express functional permissibilities // IEEE Trans. On CAD. 2000. V. 19, No. 8. P. 840–849.
6. Yamashita S., Yoshida H., Fujita M. Increasing yield using partially-programmable circuits // Proc. SASIMI. 2010. P. 237–242.

Матросова Анжела Юрьевна, д-р техн. наук, профессор. E-mail: mau11@yandex.ru

Останин Сергей Александрович, канд. техн. наук, доцент. E-mail: ostanin@mail.tsu.ru

Бухаров Александр Вениаминович. E-mail: darkhell1204@mail.ru

Кириенко Ирина Евгеньевна. E-mail: darkhell1204@mail.ru

Томский государственный университет

Поступила в редакцию 20 января 2014 г.

Matrosova Anzhela Y., Ostanin Sergei A., Bucharov Alexander V., Kirienko Irina E. (Tomsk State University, Tomsk, Russian Federation).

Generating all test patterns for a given stuck-at fault of a logical circuit and its ROBDD implementation.

Keywords: stuck-at faults; test patterns; ROBDD.

Test generation for stuck-at faults at the gate poles of a combinational circuit is reduced to finding one test pattern for each stuck-at fault. This problem is intensively investigated in Russia and abroad. Some effective methods for its solving have been developed. If we would like to get a minimal test for such faults, we need to find all test patterns for each fault and then to cover all faults with minimal subset of these test patterns. Unfortunately, this approach is very time consuming. Moreover, the number of test patterns for one fault may be very large that is why in practice, a test engineer finds any test pattern for each fault and minimize the test length using obtained subset of test patterns.

The problem of finding all test patterns for the given fault is used in many applications. It is known that the subcircuit behavior (a frame of a given circuit C) may be described with a partially specified Boolean function. This fact is used for minimizing the overall circuit as a whole and except for changing subcircuits consisting of gates with the proper CLBs during design of partially-programmable circuits and can be used for minimizing programmable arrays? Partially specified functions of subcircuits may be represented by all test patterns for corresponding stuck-at faults of the circuit. Thus, it is very important to represent all test patterns in a compact way.

This paper is devoted to solving the problem of finding all test patterns for a given stuck-at fault and compact presentation of such test patterns. The method that has been suggested by our scientific group, for finding one (any) test pattern, is modified in this paper. The modification is based on the following. We consider a stuck-at pole as the input of a subcircuit which output coincides with the output of the given circuit C . For that subcircuit the ROBDD R^c is built and the first variable of the Shannon decomposition during the ROBDD construction is the variable corresponding to the stuck-at pole. The order of other variables may be arbitrary. From this ROBDD R^c we select ROBDDs $R^{v'}$, $R^{\bar{v}'}$ which roots are high and low nodes of the ROBDD R^c root. The subcircuit of C which output is the same stuck-at pole and inputs are inputs of the circuit C is represented as ROBDD R_f . The order of variables is the same for all constructed ROBDDs. The ROBDD representing all test patterns for the given stuck-at fault is derived by multiplying and summarizing some of above mentioned ROBDDs.

First, we briefly describe a method for finding one test pattern for a given stuck-at fault based on solving the proper Boolean equations and the representation of subcircuits as the SoPs (sum of products). Then a proposed algorithm for deriving all test patterns for a given stuck-at fault is described. The method can be extended to combinational equivalent of a sequential circuit. The example illustrating the algorithm is given.

REFERENCES

1. Matrosova A.Yu. A fault-detection method for a synchronous device. *Avtomatika i telemekhanika – Automation and Remote Control*, 1977, no. 12, pp.129-137. (In Russian).
2. Matrosova A.Yu., Melnikov A.V., Mukhamedov R.V., Ostanin S.A., Singh V. Selection of the flip-flops for partial enhanced scan techniques. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*, 2012, no. 2(19), pp. 112-120. (In Russian).
3. Matrosova A.Yu., Ostanin S.A., Melnikov A.V., Singh V. Observability Calculation of State Variable Oriented to Robust PDFs and LOC or LOS Techniques. *Proc. of IEEE East-West Design&Test Symposium (EWDTS 2012)*. Ukrain, 2012, pp. 155-160.
4. Muroga S., Kambayashi Y., Lai H.C., Culliney J.N. The transduction method-design of logic networks based on permissible functions. *IEEE Trans. on Computers*, 1989. Vol. 38, no. 10, pp. 1404-1424. DOI: 10.1109/12.35836
5. Yamashita S., Sawada H., Nagoya A. SPFD: A new method to express functional permissibilities. *IEEE Trans. On CAD*, 2000. Vol. 19, No. 8. P. 840-849. DOI: 10.1109/43.856972
6. Yamashita S., Yoshida H., Fujita M. Increasing yield using partially-programmable circuits. *Proc. SASIMI*, 2010, pp. 237-242.