

На правах рукописи

Никитин Константин Владимирович

**МЕТОДЫ СИНТЕЗА САМОПРОВЕРЯЕМЫХ ДИСКРЕТНЫХ
СИСТЕМ**

05.13.01 – системный анализ, управление и обработка информации

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Томск-2003

Работа выполнена в Томском государственном университете на кафедре программирования

Научный руководитель – доктор технических наук, профессор
Матросова А.Ю.

Официальные оппоненты: доктор технических наук, профессор
Евтушенко Нина Владимировна,
кандидат технических наук, профессор
Щербанов Виктор Анатольевич

Ведущая организация Томский политехнический университет

Защита состоится «16» октября 2003 г. В 10³⁰ часов в ауд. 2126 II уч. корпус ТГУ на заседании диссертационного совета Д 212.267.12 при Томском государственном университете по адресу: 634050, г. Томск, пр. Ленина, 36.

С диссертацией можно ознакомиться в Научной библиотеке Томского государственного университета.

Автореферат разослан « ___ » сентября 2003 г.

Ученый секретарь
диссертационного совета
доктор технических наук,
профессор

Смагин В.И.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность проблемы.

Увеличивающаяся сложность и значимость дискретных (цифровых) систем требуют их высокой надежности. Сложность систем повышает вероятность возникновения неисправностей в них. С ростом уровня интеграции схем возрастает доля кратковременных и перемежающихся неисправностей по отношению к неисправностям, не меняющимся во времени. Обнаружение неисправности в первый же момент ее проявления на выходах устройства позволяет защитить систему, в которую устройство встроено, как от не меняющихся во времени неисправностей, так и перемежающихся и кратковременных. Такое обнаружение неисправности может достигаться, например, за счет использования самопроверяемых схем, позволяющих обнаруживать неисправности в режиме нормального функционирования схемы. Обнаружение осуществляется с помощью детектора кодов, который обычно является самотестируемым. Самопроверяемая схема вместе с таким детектором кодов образует полностью самопроверяемую дискретную систему.

Проектирование дискретных устройств, их производство и тестирование требуют все больших временных затрат. Подталкиваемые конкуренцией разработчики вынуждены искать пути для их сокращения. Один из способов решения данной проблемы – использование новой элементной базы, в частности, программируемых элементов и программируемых связей между элементами. Для новой элементной базы необходимы новые методы синтеза схем, учитывающие ее специфику.

Разработка методов проектирования самопроверяемых дискретных систем, ориентированных на использование новой элементной базы является, безусловно, актуальной.

Целью настоящей работы является разработка методов проектирования полностью самопроверяемых дискретных систем, ориентированных на использование программируемых логических блоков (ПЛБ). Для ее достижения решены две задачи.

1. Выполнено исследование известного метода логического синтеза комбинационной составляющей несамопроверяемого синхронного последовательностного устройства на возможность его применения к синтезу комбинационной составляющей самопроверяемого синхронного последовательностного устройства. Метод основан на покрытии системы BDD-графов, представляющей функции переходов-выходов синтезируемого синхронного автомата,

программируемыми логическими блоками (ПЛБ). Его применение при синтезе самопроверяемых устройств требует кодирование неупорядоченными кодами (в частности, равновесными) символов выходного алфавита и состояний автомата и наблюдения выходов синхронного самопроверяемого устройства и его линий обратных связей.

2. Разработан метод синтеза самотестируемых комбинационных детекторов равновесных кодов.

Поставленные задачи решаются в предположении, что каждый ПЛБ реализует либо одну (любую) булеву функцию от $k+1$ переменной, либо две (любые) булевы функции от k переменных.

Методы исследования.

В диссертационной работе используются методы дискретной математики, в частности, алгебры логики, теории автоматов и теории графов.

Научную новизну полученных в работе результатов определяют:

- Обоснование возможности применения метода логического синтеза комбинационной составляющей несамопроверяемого синхронного последовательностного устройства к синтезу комбинационной составляющей самопроверяемого синхронного последовательностного устройства. Речь идет о методе, основанном на покрытии системы BDD-графов, представляющей функции переходов-выходов автомата, программируемыми логическими блоками (ПЛБ). Его применение при синтезе самопроверяемых устройств требует кодирования символов выходного алфавита автомата и его состояний неупорядоченными кодами (например, равновесными) и наблюдения выходов самопроверяемого устройства и его линий обратных связей.
- Декомпозиционный метод проектирования самотестируемых комбинационных детекторов равновесных кодов в базисе ПЛБ. Детектор кодов является самотестируемым относительно кратных константных неисправностей на полюсах ПЛБ. Проблема синтеза самотестируемых детекторов в базисе ПЛБ исследуется впервые.
- Алгоритмы сокращения числа допустимых кодовых слов на входах самотестируемых детекторов.
- Формула оценки сложности детектора кодов, то есть числа ПЛБ, необходимых для реализации детектора.

Достоверность полученных результатов.

Все научные положения и выводы, содержащиеся в диссертации, доказаны с использованием аппарата дискретной математики. Эффективность предложенных методов синтеза подтверждена компьютерными экспериментами и теоретическими расчетами.

Практическая значимость работы.

Предложенные в работе методы синтеза самопроверяемых синхронных последовательностных устройств и самотестируемых комбинационных детекторов равновесных кодов могут быть применены на этапе логического проектирования дискретных устройств с целью создания самопроверяемых систем. Важно, что для этого не требуется специальных средств автоматизированного логического проектирования, достаточно уже существующих. Синтезированные этим методом устройства являются полностью самопроверяемыми в классе одиночных константных неисправностей на входах и выходах, а так же одиночных функциональных неисправностей ПЛБ для проверяемого устройства и в классе кратных константных неисправностей на полюсах ПЛБ для детектора кодов. Данное множество неисправностей является довольно широким и покрывает большую часть реально возникающих неисправностей.

Разработанные методы синтеза самопроверяемых устройств ориентированы на реализацию в базисе программируемых элементов, в частности ПЛБ Xilinx 3000. Проектирование может также быть выполнено в базисе ПЛБ любых производителей, удовлетворяющих описанным ранее требованиям, предъявляемым к ПЛБ.

Полученная формула подсчета числа ПЛБ позволяет оценить сложность детектора до проектирования устройства и, следовательно, выяснить целесообразность такого проектирования.

Реализация полученных результатов.

Исследования, результаты которых изложены в диссертации, проводились в рамках следующих проектов.

1. Госбюджетная тема Сибирского физико-технического института при ТГУ, программа “Исследование и разработка новых методов электромагнитного контроля и диагностики материалов, сред и технических систем”, 1995-2000 гг., раздел “Разработка методик и аппаратуры исследований”.
2. Межвузовская научно-техническая программа “Конверсия и высокие технологии. 1994-2000 гг.”, проект №95-1-21 и №59-1-7 “Информационные компьютерные технологии дискретного

математического моделирования, анализа, синтеза и тестирования сверхскоростных интегральных схем логического управления”.

3. Научный проект Минобразования России «Решение логических уравнений на BDD-графах в задачах диагностики».

Результаты работы также используются в курсе лекций “Диагностика дискретных устройств” на факультете прикладной математики и кибернетики Томского государственного университета (ТГУ).

Апробация работы и публикации.

Научные результаты, составляющие основу данной работы, по мере их получения обсуждались на заседаниях объединенного семинара кафедры математической логики и проектирования радиофизического факультета ТГУ, кафедры программирования, кафедры защиты информации факультета прикладной математики и кибернетики ТГУ и лаборатории синтеза дискретных автоматов Сибирского физико-технического института (СФТИ) при ТГУ.

Результаты работы представлялись на следующих научных конференциях:

1. Третья всероссийская конференция с международным участием «Новые информационные технологии в исследовании дискретных структур» (Россия, Томск 2000);
2. The Fourth International Conference on Computer-added Design of Discrete Devices (CAD DD'2001) (Minsk, Republic of Belarus, November 14-16, 2001);
3. Международная конференция «Компьютерные науки и информационные технологии» (Россия, Саратов 14-18 мая 2002г.).
4. 7th IEEE International On-Line Testing Workshop (Taormina, Italy, July 9-11, 2001);
5. 9th IEEE International On-Line Testing Workshop, Greece, Kos, July 7-9, 2003.

По результатам выполненных исследований опубликовано 7 печатных работ.

Структура и объем диссертации.

Диссертация состоит из введения, 4 глав, заключения и списка используемой литературы. Диссертация содержит 33 рисунка и 14 таблиц. Объем диссертации составляет 115 стр., в том числе: титульный лист - 1 стр., оглавление - 2 стр., основной текст - 104 стр., библиография из 90 наименования - 8 стр.

СОДЕРЖАНИЕ РАБОТЫ

Во введении дается обоснование актуальности работы, изложены ее цель, научная новизна и практическая ценность.

Первая глава содержит основные определения: булевых функций, конечных автоматов, логических схем и элементов логических схем.

Определение 1. Функция $f(x_1, x_2, \dots, x_n)$, $x_i \in E^2 = \{0, 1\}$, $i = \overline{1, n}$, причем, $f(\alpha_1, \alpha_2, \dots, \alpha_n) \in E^2$, где $\alpha_i \in E^2$, $i = \overline{1, n}$ называется *булевой функцией*.

Определение 2. Автомат – это пятерка (X, Q, Y, ψ, φ) , где X, Q, Y – конечные алфавиты, называемые входным алфавитом, множеством состояний и выходным алфавитом. ψ – функция переходов, отображающая множество $X \times Q$ в Q , φ – функция выходов, отображающая множество $X \times Q$ в Y .

Для минимизации времени, необходимого на проектирование и производство схем, используются программируемые пользователем большие интегральные схемы, которые обладают наибольшей гибкостью использования и оперативностью. Компоненты такого аппаратного оборудования (логические блоки и связи) находятся на уже произведенном чипе и могут быть запрограммированы, чтобы реализовать любой вид цифровой схемы.

В последние годы большое развитие получила FPGA (Field of Programmable Gate Array) технология. Основу FPGA технологии составляют программируемые логические блоки – CLB (Configurable Logic Block). CLB – это конфигурация логических элементов, которые могут быть программируемы пользователем. Существуют реализации CLB (в частности Xilinx 3000), где каждый CLB может выполнять одну или две (любые) булевы функции от заданного числа переменных. Если CLB реализует две функции, то эти функции могут зависеть от одних и тех же переменных.

Поскольку исследования диссертационной работы не ориентированы на конкретную реализацию FPGA технологии, в работе введено понятие программируемого логического блока (ПЛБ). Каждый ПЛБ имеет $k+1$ вход и один выход или k входов и два выхода. Если ПЛБ имеет один выход, то на этом выходе может быть реализована одна (любая) булева функция, зависящая от $k+1$ переменных. В случае если ПЛБ имеет два выхода, то он может выполнять любые две булевы функции, каждая из которых зависит от одних и тех же k входных переменных.

Существуют два основных подхода к проверке исправности работы схемы:

- метод тестового диагностирования;
- метод функционального диагностирования.

При тестовом диагностировании для проверки исправности работы схема переводится в тестовый режим, на ее входы подаются специальные тестовые наборы, и анализируется реакция схемы на эти наборы. Для получения тестовых наборов используется генератор тестов, ориентированный на обнаружение неисправностей из рассматриваемого множества. Генератор тестов реализуется специальным (дополнительным) оборудованием.

Метод функционального диагностирования не предусматривает перевода схемы в специальный режим, исправность схемы относительно некоторого рассматриваемого множества неисправностей определяется в процессе нормального функционирования схемы. При реализации данного метода выходные наборы схемы представляют собой кодовые слова некоторого кода (например, равновесного кода, кода Бергера и т.д.). Специальное кодирование выходных наборов приводит к увеличению количества выходов схемы. К выходам схемы должно постоянно быть подключено проверяющее устройство – детектор кодов. Если некоторый выходной набор схемы не является кодовым словом, то детектор кодов сигнализирует об этом. В этом случае оборудование, обеспечивающее тестовое диагностирование, размещается непосредственно на том же чипе, где размещена основная схема (тестируемая схема). В рассматриваемом подходе дополнительным оборудованием являются дополнительные выходы схемы, вместе с реализуемыми ими функциями, и детектор кодов.

В данной работе рассматривается метод функционального диагностирования. Если при появлении неисправности из некоторого класса неисправностей в процессе нормального функционирования схемы (в рабочей области функционирования) данная неисправность либо обнаружится на выходах схемы, либо никак не влияет на функционировании схемы (остается необнаружимой), то такую схему будем называть *самопроверяемой* относительно рассматриваемого класса неисправностей.

Если в рабочей области функционирования схемы детектора неисправность из некоторого класса обязательно проявится на выходах детектора кодов, то такой детектор кодов будем называть *самотестируемым* относительно рассматриваемого класса неисправностей.

В первой главе также проводится обзор известных методов проектирования самопроверяемых дискретных устройств и самотестируемых детекторов кодов.

Во второй главе устанавливается, что подход к синтезу синхронной последовательностной схемы, основанный на описании функционирования схемы BDD-графами и последующем покрытии этого описания программируемыми логическими блоками (ПЛБ) может быть также применен к синтезу самопроверяемых схем при условии соответствующего кодирования символов выходного алфавита и состояний автомата.

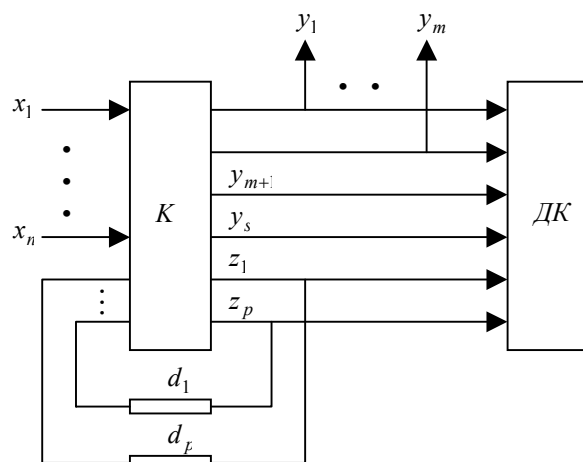


Рис. 1. Самопроверяемая синхронная последовательностная схема

Рассмотрим схему, представленную на рис. 1. В ней комбинационная часть синхронного автомата реализует следующие функции: функции выходов автомата (y_1, \dots, y_m) , функции дополнительных выходов (y_{m+1}, \dots, y_s) , обеспечивающие специальное кодирование выходов схемы, и функции переходов для представления следующего состояния (z_1, \dots, z_p) . Выходы y_1, \dots, y_s формируют кодовые слова равновесного или любого неупорядоченного кода.

На рис. 1. K является комбинационной частью, d_1, \dots, d_p – задержки, реализуемые d -триггерами.

Комбинационная схема K реализует систему F из $(s + p)$ булевых функций, зависящих от $(n + p)$ булевых переменных. В случае возникновения неисправности w , комбинационная схема K реализует систему булевых функций F^w .

Пусть α_1, α_2 – входные наборы, представленные булевыми векторами длины $(n + p)$. Если для любых компонент $\alpha_{1i}, \alpha_{2i}, i = \overline{1, (n + p)}$, $\alpha_{1i} \leq \alpha_{2i}$, то $\alpha_1 \leq \alpha_2$. Например, если $\alpha_1 = 010010$, $\alpha_2 = 110110$, то $\alpha_1 \leq \alpha_2$.

$F(\alpha)$ – булев вектор длины $(s + p)$, $F(\alpha) = \beta$. Он представляет значение системы F функций на входном наборе α : $F(\alpha_1) = \beta_1$, $F(\alpha_2) = \beta_2$.

Пусть $\tilde{\alpha}_1, \tilde{\alpha}_2$ – различные последовательности булевых векторов. Последовательности $\tilde{\alpha}_1, \tilde{\alpha}_2$ сравнимы, если для любой пары $\alpha_1^i, \alpha_2^i, i \in \{1, 2, \dots\}$ их элементов имеем: $\alpha_1^i \leq \alpha_2^i$ или $\alpha_2^i \leq \alpha_1^i$. Обозначим их как $\tilde{\alpha}_1 \lesseqgtr \tilde{\alpha}_2$. Например, последовательности $\tilde{\alpha}_1 = 010010, 011010, 111011$ и $\tilde{\alpha}_2 = 010010, 001010, 111111$ сравнимы.

Пусть неисправность w проявляет себя как однонаправленная неисправность. Однонаправленная неисправность вызывает замену в выходном векторе β либо только нулей на единицы либо только единиц на нули, но не те и другие одновременно. Если набор α является тестом для некоторой однонаправленной неисправности w схемы K , то имеем: либо $F(\alpha) < F^w(\alpha)$, либо $F^w(\alpha) < F(\alpha)$. Если входной набор α не является тестом, то $F(\alpha) = F^w(\alpha)$.

Пусть β, β^w равны $F(\alpha), F^w(\alpha)$, соответственно, и $\tilde{\beta} = F(\tilde{\alpha}), \tilde{\beta}^w = F^w(\tilde{\alpha})$.

Теорема 1. Если w является однонаправленной неисправностью, то для любой последовательности $\tilde{\alpha}$ $F(\tilde{\alpha}) \lesseqgtr F^w(\tilde{\alpha})$ то есть $\tilde{\beta} \lesseqgtr \tilde{\beta}^w$.

Пусть F есть система из m булевых функций, зависящих от n булевых переменных x_1, \dots, x_n , $\{x_1, \dots, x_n\} = X$.

Определение 3. Система F функций является монотонной, если для любой пары булевых векторов α_1, α_2 , таких что $\alpha_1 \leq \alpha_2$, имеет место условие: $F(\alpha_1) \leq F(\alpha_2)$, то есть $\beta_1 \leq \beta_2$, где $\beta_1 = F(\alpha_1)$, $\beta_2 = F(\alpha_2)$.

Рассмотрим подмножество X^* булевых переменных, $X^* = \{x_{i_1}, \dots, x_{i_r}\}$, $X^* \subset X$. Пусть α_1, α_2 – булевы векторы n переменных.

Определение 4. Если для любого i , соответствующего переменной x_i из множества X^* , $\alpha_{1i} \leq \alpha_{2i}$, и для любого j , соответствующего переменной x_j , $x_j \in \{X \setminus X^*\}$, $\alpha_{1j} = \alpha_{2j}$, то $\alpha_1 \leq \alpha_2$.

Определение 5. Система F функций является частично монотонной по переменным множества X^* , если для любой пары булевых векторов α_1, α_2 , таких что $\alpha_1 \leq \alpha_2$, имеет место условие: $F(\alpha_1) \leq F(\alpha_2)$, ($\beta_1 \leq \beta_2$, $\beta_1 = F(\alpha_1)$, $\beta_2 = F(\alpha_2)$).

Определение 6. Неисправность называется однонаправленной, если ее проявление на наблюдаемых выходах приводит к замене значений некоторых переменных только с «0» на «1» или с «1» на «0», но не в обоих направлениях одновременно.

Определение 7. BDD-графом булевой функции называется ориентированный граф с корнем, имеющий множество вершин V . V содержит два типа вершин: нетерминальные и терминальные. Нетерминальная вершина v , $v \in V$, имеет в качестве атрибутов индекс аргумента $index(v) \in \{1, \dots, n\}$ и две дочерние вершины $high(v), low(v) \in V$. Терминальная вершина v имеет один атрибут – $value(v) \in \{0, 1\}$.

Функция f_v , соответствующая вершине v BDD – графа, определяется рекурсивно следующим образом:

1. Если v является терминальной вершиной, и
 - a. $value(v) = 1$, то $f_v = 1$,
 - b. $value(v) = 0$, то $f_v = 0$.
2. Если v – нетерминальная вершина с индексом $index(v) = i$, то f_v является функцией следующего вида:

$$f_v(x_1, \dots, x_n) = \bar{x}_i \cdot f_{low(v)}(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \vee x_i \cdot f_{high(v)}(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

Одной из важнейших операций манипулирования BDD-графами является сокращение графа. Граф может быть сокращен без изменения представляемой им функции путем удаления лишних вершин и совпадающих подграфов с целью получения его канонической формы при выбранном порядке разложения по переменным.

Обычно поведение синхронного последовательностного устройства задается либо таблицами переходов – выходов, либо микропрограммным описанием. Воспользуемся STG-описанием, пример которого представлен в

Табл. 1, аналогичным микропрограммному. Здесь символы входного и выходного алфавита уже закодированы. После кодирования состояний получаем интервальное описание (Табл. 2). Для обеспечения самопроверяемости состояния будем кодировать кодовыми словами какого либо неупорядоченного кода, например равновесного.

Имея интервальное описание системы булевых функций переходов-выходов автомата, можно получить представление комбинационной схемы автомата в виде системы BDD-графов, а затем покрыть полученную систему программируемыми логическими блоками, с тем, чтобы получить комбинационную схему K в базисе ПЛБ.

Таблица 1.

$x_1x_2x_3$	Q	Q	$y_1y_2y_3y_4y_5$
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.

$x_1x_2x_3$	$z_1z_2z_3z_4$	$z_1z_2z_3z_4$	$y_1y_2y_3y_4y_5$
0 - -	1 0 0 0	1 0 0 0	0 0 0 1 0
- 0 -	1 0 0 0	1 0 0 0	0 0 0 1 0
1 1 -	1 0 0 0	0 1 0 0	1 0 0 1 0
- - 0	0 1 0 0	0 1 0 0	0 0 1 1 0
- - 1	0 1 0 0	0 0 1 0	1 0 1 1 0
1 0 -	0 0 1 0	0 0 1 0	0 1 0 0 0
0 - -	0 0 1 0	0 0 0 1	1 1 0 0 0
- 1 -	0 0 1 0	0 0 0 1	1 1 0 0 0
- - 0	0 0 0 1	0 0 0 1	0 1 0 0 1
- - 1	0 0 0 1	1 0 0 0	1 1 0 0 1

Предложен алгоритм покрытия системы BDD-графов программируемыми логическими элементами (ПЛБ).

1. Выполняем обход графа от терминальных вершин к корню. Для каждой вершины v определяем множество B_v переменных, от которых зависит функция, представленная графом с корнем в вершине v . В результате получаем граф, все вершины которого размечены множествами B_v . Назовем его *размеченным графом*.
2. Выбираем любую вершину u , мощность множества B_u которой равна k . Если такой вершины нет, то ищем вершину, для которой мощность множества B_u равна $(k-1)$ и т.д., пока не определим вершину u .

3. Сопоставляем вершине u ПЛБ, который реализует функцию, представленную графом с корнем в вершине u . Сама вершина u сопоставляется новой переменной – w_i .
4. Выполняем п.1 данного алгоритма для новой системы BDD-графов и т.д., пока не покроем систему BDD-графов целиком.

Введем следующий класс неисправностей W : одиночные константные неисправности на входных полюсах синхронной последовательностной схемы, одиночные константные неисправности на ее выходных полюсах и полюсах d -триггеров и одиночные функциональные неисправности функций, реализуемых ПЛБ. На выходе ПЛБ в случае возникновения такой неисправности может быть реализована любая другая функция того же множества переменных. В синхронной последовательностной схеме возможна единственная неисправность из W .

Рассмотрим функциональную неисправность w . Соответствующий ПЛБ реализует вместо φ функцию φ^w , зависящую от входных переменных ПЛБ. Пусть $\tilde{\alpha}$ – произвольная входная последовательность схемы K и F^w – неисправная система, которую реализует схема в присутствии рассматриваемой неисправности.

Теорема 2. $F(\tilde{\alpha}) \leq F^w(\tilde{\alpha})$.

Следствие. Любая одиночная функциональная неисправность проявляется как однонаправленная неисправность.

Рассмотрим одиночную неисправность w_d , $w_d \in W$, на полюсе d -триггера. Любая такая неисправность проявляется как одиночная неисправность на входе K , соответствующей линии обратных связей синхронного последовательностного устройства.

Теорема 3. $F^{w_d}(\tilde{\alpha}) \leq F(\tilde{\alpha})$.

Следствие. Любая неисправность w_d проявляется как однонаправленная неисправность.

Рассмотрим одиночную константную неисправность w_i , $w_i \in W$, одной из входных переменных синхронной последовательностной схемы.

Теорема 4. $F^{w_i}(\tilde{\alpha}) \leq F(\tilde{\alpha})$.

Следствие 1. Любая неисправность w_i проявляется как однонаправленная неисправность.

Следствие 2. Любое покрытие программируемыми логическими блоками (ПЛБ) BDD-графов сохраняет свойства однонаправленного проявления неисправностей множества W .

Так как любая одиночная константная неисправность на выходе K очевидно является однонаправленной, то в соответствии со следствиями теорем 2, 3, 4, любая неисправность $w \in W$ является однонаправленной.

Если исключить неисправности на входах, то нет необходимости в добавлении входных переменных. Такое сокращенное множество неисправностей тоже используется при синтезе самопроверяемых схем. В этом случае ограничимся кодированием состояний и выходов, не вводя дополнительные входные переменные.

Покрытие системы BDD-графов программируемыми логическими блоками допускает различные модификации, ориентированные на оптимизацию структуры синхронной последовательностной схемы. При этом любое покрытие системы BDD-графов сохраняет свойства однонаправленного проявления неисправностей из множества W .

В третьей главе предлагается метод проектирования самотестируемого детектора (m,n) -кодов, ориентированный на использование программируемых логических блоков (ПЛБ).

Рассматривается множество неисправностей детектора V , которое включает в себя все кратные константные неисправности на входах и выходах ПЛБ. Только один ПЛБ в детекторе может быть неисправен.

К детектору предъявляются следующие требования.

При появлении в момент времени t на выходе схемы некодового слова, детектор в тот же момент t выдает сигнал, свидетельствующий о неисправности схемы, к выходам которой он подключен. В самом детекторе может произойти неисправность из рассматриваемого множества неисправностей V , которая должна быть обнаружима в рабочей области функционирования детектора. Это значит, что на выходах самопроверяемой исправной схемы должно существовать кодовое слово (m,n) -кода, на котором эта неисправность проявляется.

Возможна либо неисправность самопроверяемой схемы, либо самотестируемого детектора.

Самотестируемый детектор имеет два выхода (рис. 2). Комбинации значений сигналов:

а) (01) или (10) означают, что выходной набор самопроверяемой схемы является кодовым словом и детектор кодов исправен;

б) (00) или (11) означают, что либо выходной набор самопроверяемой схемы не является кодовым словом, либо детектор неисправен.

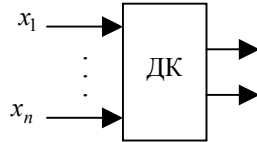


Рис.2. Детектор кодов

Пусть M – множество кодовых слов (m, n) -кода, F^v – система двух булевых функций, реализуемых детектором в присутствии неисправности из рассматриваемого множества неисправностей V . Тогда для всякой неисправности $v, v \in V$ существует тестовый набор $\alpha, \alpha \in M$ такой, что неисправность монотонно проявляет себя на выходах детектора, т.е. либо $F(\alpha) < F^v(\alpha)$, либо $F^v(\alpha) < F(\alpha)$.

Заметим, что число всевозможных кодовых слов (m, n) -кода равно C_n^m , то есть числу сочетаний из n по m . Кодовые слова могут быть представлены дизъюнкцией конъюнкций ранга n . Обозначим эту дизъюнкцию $D_m^n(X)$, где $X = \{x_1, \dots, x_n\}$ – множество переменных. Уже при $n = 10, m = 5$ $D_{10}^5(X)$ состоит из 252 конъюнкций ранга 10 и содержит 2520 букв. Это выражение не может быть сокращено в результате минимизации ДНФ, ДНФ $D_m^n(X)$ является совершенной и сокращенной одновременно, поскольку любые две конъюнкции из $D_m^n(X)$ ортогональны по крайней мере по двум переменным.

Для представления всевозможных (m, n) -кодов предложена специальная формула разложения A . Формула включает скобки, символы \wedge, \vee и ДНФ $D_p^q(X^r)$. Функция разложения $D_p^q(X^r)$ – это дизъюнкция конъюнкций, соответствующих всем (q, p) -кодам, $p \leq k, q \leq p, X^r \subset X, X = \{x_1, \dots, x_n\}$. В частности, k может быть равно числу входов двухвыходного ПЛБ. Каждый ПЛБ либо реализует одну произвольную булеву функцию от $k+1$ переменных, либо две произвольные булевы функции от k переменных, и D_p^q – функция, сопоставляемая одному из выходов ПЛБ.

Разделим множество X на два подмножества X^1, X^* , где $X^1 = \{x_1, \dots, x_k\}, X^* = \{x_{k+1}, \dots, x_n\}$.

Теорема 5.

$$(1) \quad D_n^m(X) = D_k^0(X^1)D_{n-k}^m(X^*) \vee D_k^1(X^1)D_{n-k}^{m-1}(X^*) \vee \dots \vee D_k^k(X^1)D_{n-k}^{m-k}(X^*)$$

Здесь $D_n^m(X)$ – дизъюнкция конъюнкций, представляющих всевозможные кодовые слова (m, n) -кода. Назовем $D_k^i(X^1)$, $D_{n-k}^j(X^*)$ функциями разложения. Если $n-k > k$, то выполним следующий шаг данного разложения для каждой $D_{n-k}^j(X^*)$, и так далее. В результате получим формулу A , в которой для любой $D_p^q(X^r)$ выполняется условие $p \leq k$, где k равно числу входов ПЛБ. Данное разложение обозначается формулой A .

Теорема 6. Число l различных конъюнкций любого выражения (1) не более $k + 1$.

Представим выражение (1) *деревом разложения* (рис. 3). Корень дерева пометим символом \vee , а l смежных с ним вершин – символом \wedge . Его листьям соответствуют функциям разложения. Корню дерева и его l неконцевым вершинам сопоставляются булевы функции, являющиеся композицией функций разложения. Каждая из этих функций зависит от одного и того же подмножества переменных формулы A .

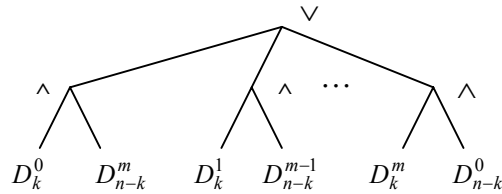


Рис. 3. Дерево разложения

Если на некотором наборе значений этих переменных функция, сопоставляемая неконцевой вершине дерева, обращается в единицу, то будем говорить, что вершина дерева *активизирована*. Дерево разложения будем называть *активизированным*, если активизирована его корневая вершина. Последнее имеет место, если активизирована хотя бы одна из вершин, помеченных символом \wedge .

Теорема 7. Любое кодовое слово некоторого (m, n) -кода l обращает в единицу \vee функций разложения, каждая из которых зависит не более чем от

k переменных, 2) активизирует $\nu-1$ деревьев разложения и ровно одну вершину, отмеченную символом " \wedge ", в каждом активизированном дереве разложения, причем, 3) корни активизированных вершин соединены между собой и корнем дерева A единственной простой цепью.

Следствие. Любое кодовое слово (m,n) -кода активизирует в любом дереве разложения формулы A не более одной вершины, отмеченной символом " \wedge ".

Теорема 8. Для любого дерева разложения и любой его вершины, отмеченной символом \wedge , существует набор, являющийся кодовым словом (m,n) -кода, активизирующий эту вершину.

Все кодовые слова (m,n) -кода разделяются на два подмножества, и формула A представляется следующим образом: $A = A^1 \vee A^2$. Так как каждая конъюнкция ДНФ D_n^m , представляющей все кодовые слова (m,n) -кода, ортогональна любой другой конъюнкции этой ДНФ, то можно представлять формулу A следующим образом: $A = A_1 \oplus A_2$. Здесь A^1 реализует кодовые слова первого подмножества, A^2 – оставшиеся кодовые слова (m,n) -кода. A^1 и A^2 соответствуют двум выходам детектора.

Построим двухвыходную схему C (детектор (m,n) - кода), которая реализует на своих выходах формулы A_1 и A_2 . Значения 10, 01 достигаются на выходах C , если на ее входы поступают кодовые слова (m,n) -кода, значения 00, 11 – если на входы поступают некодовые слова.

Предложен алгоритм покрытия схемы ПЛБ. При разработке данного алгоритма учтена специфика ПЛБ, а именно возможность реализации ПЛБ любой булевой функции от заданного числа переменных, а также возможность реализации двух функций одним ПЛБ.

Пусть переменные $y_1, y_2, \dots, y_{2l-1}, y_{2l}$ сопоставлены различным функциям разложения в выражении (1). Тогда его правая часть представляется в виде:

$$(2) \quad y_1 y_2 \vee y_3 y_4 \vee \dots \vee y_{2l-1} y_{2l}.$$

Согласно следствию теоремы 7 любое кодовое слово обращает в единицу не более одной конъюнкции выражения (2). Это позволяет заменить выражение (2) дизъюнкцией конъюнкций, представляющих $(2,2l)$ -коды.

Например, выражение $y_1 y_2 \vee y_3 y_4 \vee y_5 y_6$ можно представить в виде: $y_1 y_2 \bar{y}_3 \bar{y}_4 \bar{y}_5 \bar{y}_6 \vee \bar{y}_1 \bar{y}_2 y_3 y_4 \bar{y}_5 \bar{y}_6 \vee \bar{y}_1 \bar{y}_2 \bar{y}_3 \bar{y}_4 y_5 y_6$.

Реализуем выражение (2) одновыходными ПЛБ с $k+1$ входами. Пусть k нечетно. Выполним следующие шаги.

1. Выделим первые $\frac{k+1}{2}$ конъюнкций выражения (2) с последующим изменением их на дизъюнкцию конъюнкций, представляющих соответствующие $(2, k+1)$ -коды. Блок ПЛБ₁ реализует эту дизъюнкцию конъюнкций. Если число конъюнкций выражения (2) меньше или равно $\frac{k+1}{2}$, тогда это выражение реализуется единственным одновыходным ПЛБ₁. В противном случае выполняем шаг 2.

2. Возможны следующие ситуации.

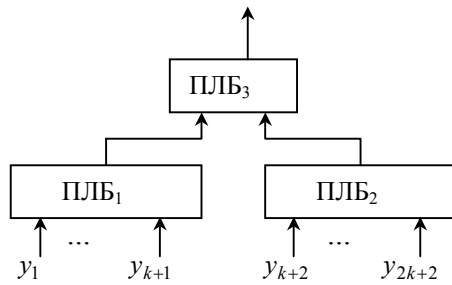


Рис. 4. Подсхема D для $l = k+1, \nu > 1$.

а) Число оставшихся конъюнкций выражения (2) равно $\frac{k+1}{2}$ (согласно теореме 6 это число конъюнкций является максимальным). Используем одновыходной ПЛБ₂ для реализации этих конъюнкций в виде суммы конъюнкций, представляющих соответствующие $(2, k+1)$ -коды.

Для реализации выражения (2) в целом необходим еще один блок. Подсхема D , реализующая выражение (2), представлена на рис. 4.

б) Число оставшихся конъюнкций выражения (2) меньше или равно $\frac{k-1}{2}$. Соответствующая подсхема D представлена на рис. 5.

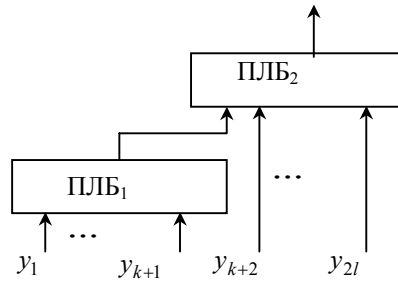


Рис. 5. Подсхема D для $l \leq k, v > 1$

Теорема 9. Схема является самотестируемой для множества V неисправностей.

В таблице 3 приведены результаты сравнения с другими известными в литературе реализациями детекторов (m, n) -кодов. Здесь первая строка соответствует предлагаемому методу синтеза детекторов, вторая – детекторам, описанным в зарубежных публикациях при условии покрытия предложенных в них схем программируемыми логическими блоками (ПЛБ). Число ПЛБ в предлагаемом методе, как правило, меньше. Следует иметь в виду, что в зарубежных публикациях рассматриваются методы синтеза, ориентированные прежде всего на вентиляльную реализацию и обеспечение самотестируемости для одиночных константных неисправностей на полюсах вентилях. Предлагаемый метод ориентирован на реализацию в рамках ПЛБ и более широкий класс неисправностей.

Таблица 3

	D_5^2	D_6^2	D_7^2	D_9^2	D_6^3	D_8^4	D_{10}^5	D_{12}^6
ПЛБ в предложенном методе	2	6	6	8	6	9	18	25
Другие методы при условии покрытия ПЛБ	7	10			5	14	24	36

В четвертой главе предлагаются оценки сложности детекторов, описанных в главе 3. Выведены формулы для подсчета числа ПЛБ, необходимых для реализации самотестируемого детектора при следующих предположениях относительно параметров детектора:

- а) рассматривается детектор $(2n, n)$ -кодов;

б) число входов ПЛБ равно k , и n делится на k , $\frac{n}{k} \geq 1$.

Полученные формулы (3) позволяют подсчитать число ПЛБ при различных значениях k , и n для четного или нечетного числа входов ПЛБ.

$$(3) \quad \begin{aligned} N_{чет} &= 3 \cdot \left[\frac{n-k+2}{2} \cdot \frac{n}{k} + \frac{n+2k+2}{2} \cdot \left(\frac{n}{k} - 1 \right) \right] - \left(\frac{n}{k} - 1 \right) \cdot (3k+2) + \frac{n}{k} \cdot (k+2) \\ N_{неч} &= 3 \cdot \left[\frac{n-k+2}{2} \cdot \frac{n}{k} + \frac{n+2k+2}{2} \cdot \left(\frac{n}{k} - 1 \right) \right] - \left(\frac{n}{k} - 1 \right) \cdot (3k+3) + \frac{n}{k} \cdot (k+1) \end{aligned}$$

Из формул следует, что сложность детектора растет пропорционально квадрату длины его кодовых слов (квадрату n). Эти формулы могут служить оценкой сложности детекторов (m, n) -кодов в широком диапазоне значений m и n .

Обычно свойства самотестируемости детектора доказываются в предположении, что все кодовые слова некоторого равновесного кода достигаются на его входах. Множество этих кодовых слов будем называть допустимым на входах детектора. На практике это не всегда возможно: число кодовых слов на выходах самопроверяемой схемы меньше числа всевозможных кодовых слов равновесного кода. В этом случае желательно обеспечение свойства самотестируемости на подмножестве кодовых слов, близком к подмножеству кодовых слов, достижимых на выходах самопроверяемой схемы.

Один из способов решения этой проблемы – построение детекторов неупорядоченных кодов на основе детекторов различных (m, n) -кодов.

Доказано, что при использовании неупорядоченного кода специального вида (4) детектор кода остается самотестируемым при сокращении числа допустимых кодовых слов по сравнению с подходящим детектором (m, n) -кодов. При этом число ПЛБ, необходимых для реализации детектора неупорядоченных кодов, не увеличивается.

$$(4) \quad D_{n-1}^{m_1} \cdot \bar{x} \vee D_{n-1}^{m_2} \cdot x, \quad m_1 - m_2 > 1$$

Второй подход заключается в сокращении числа кодовых слов (m, n) -кода, достижимых на входе детектора путем исключения в формуле A подформул. Будем удалять в дереве формулы A поддеревья, так чтобы оставшееся дерево представляло кодовые слова (m, n) -кода.

1. Удаляем только поддеревья, корни которых отмеченные символом \wedge .
2. Поддерево удаляем только в том случае, если у следующей вершины, отмеченной \vee (при движении по дереву к корню),

кроме вершины, являющейся корнем удаляемого поддеревя, существует не менее одной дочерней вершины, отмеченных символом \wedge .

Самотестируемость детектора в классе рассматриваемых неисправностей сохраняется.

Основные положения, выдвигаемые на защиту.

1. Обоснование возможности применения метода логического синтеза комбинационной составляющей несамопроверяемого синхронного последовательностного устройства к синтезу комбинационной составляющей самопроверяемого синхронного последовательностного устройства. Речь идет о методе покрытия программируемыми логическими блоками (ПЛБ) системы BDD-графов, представляющей функции переходов-выходов автомата. Состояния и символы выходного алфавита синхронного автомата кодируются при синтезе самопроверяемого устройства неупорядоченными кодами. Наблюдаются выходы и линии обратных связей устройства. Допускаются функциональные неисправности ПЛБ и одиночные константные неисправности на полюсах триггеров и входных полюсах.

2. Метод проектирования самотестируемых комбинационных детекторов равновесных кодов, ориентированный на использование ПЛБ. Допускаются кратные неисправности на полюсах одного ПЛБ. Проблема синтеза детекторов такого типа рассмотрена впервые.

3. Формула подсчета числа ПЛБ, необходимых для реализации детектора равновесных кодов.

4. Алгоритмы сокращения числа допустимых кодовых слов, поступающих на вход детектора, при сохранении свойств самотестируемости детектора.

Основные результаты диссертации опубликованы в работах:

1. Матросова А.Ю., Никитин К.В. Проектирование самотестируемого детектора неупорядоченных кодов // Докл. Международной конференции Компьютерные науки и информационные технологии. Саратов, 2002. – С. 44-45.

2. A. Matrosova, O. Goloubeva, K. Nikitin, S. Ostanin. Self-Checking FSM Design Based on BDD Synthesis methods and FPGA Implementation // The Fourth International Conference on Computer-aided Design of Discrete Devices, Minsk, Republic of Belarus, November 14-16, 2001 – P. 23-31.

3. К.В. Никитин. Об оценке сложности комбинационного детектора равновесных кодов // Третья всероссийская конференция с международным

участием Новые информационные технологии в исследовании дискретных структур, Томск, 2000 – С. 252-256.

4. Матросова А.Ю., Никитин К.В. Синтез самопроверяемого детектора равновесных кодов // Вестник Томского государственного университета № 271 – 2000. – С. 101-105.

5. A. Matrosova, K. Nikitin, O. Goloubeva. Totally self-checking FSM design based on multilevel synthesis methods and FPGA implementation // 7th IEEE International On-Line Testing Workshop, Taormina, Italy – July 9-11, 2001 – P. 144.

6. А.Ю. Матросова, К.В.Никитин. Синтез самотестируемого детектора (m,n) – кодов на программируемых логических блоках // Вестник Томского государственного университета. Приложение. №6. 2003. – С. 124-136.

7. A. Matrosova, V. Ostrovsky, I. Levin, K. Nikitin. Designing FPGA based Self-Testing Checkers for m-out-of-n Codes // 9th IEEE International On-Line Testing Workshop, Greece, Kos. – July 7-9, 2003 – P. 49-53.