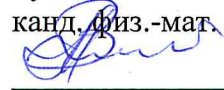


Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Руководитель ООП

канд. физ.-мат. наук, доцент



В.А. Мещеряков

« 24 » августа 2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА
(ДИПЛОМНАЯ РАБОТА)

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА ПОЛНОТЫ ПРОВЕРЯЮЩЕГО ТЕСТА,
ПОСТРОЕННОГО ДЛЯ ОБНАРУЖЕНИЯ ВЫХОДНЫХ ОШИБОК В КОНЕЧНОМ
АВТОМАТЕ, ОТНОСИТЕЛЬНО ОДИНОЧНЫХ НЕИСПРАВНОСТЕЙ В ЛОГИЧЕСКИХ
СХЕМАХ

по специальности 11.05.01 «Радиоэлектронные системы и комплексы»
специализация «Радиоэлектронные системы передачи информации»

Карепин Эдуард Денисович

Руководитель ВКР

канд. техн. наук, доцент



С.А. Прокопенко

« 24 » 01 2024 г.

Автор работы

студент группы № 07809



Э.Д. Карепин

« 24 » 01 2024 г.

Томск – 2024

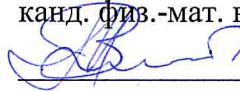
Министерство науки и высшего образования Российской Федерации.

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Радиофизический факультет

УТВЕРЖДАЮ

Руководитель ООП

канд. физ.-мат. наук, доцент

 В.А. Мещеряков

«28» апреля 2023 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы специалиста обучающемуся Карепину Эдуарду Денисовичу
по направлению подготовки 11.05.01 «Радиоэлектронные системы и комплексы»,
направленность (профиль) «Радиоэлектронные системы передачи информации»

1 Тема выпускной квалификационной работы

Экспериментальная оценка полноты проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно одиночных неисправностей в логических схемах

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:

- а) в учебный офис / деканат – 24.124 ;
б) в ГЭК – 24.1.24 .

3 Исходные данные к работе:

Объект исследования – проверяющий тест, построенный для обнаружения выходных ошибок в конечном автомате.

Предмет исследования – полнота проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате.

Цель исследования – экспериментальная оценка полноты проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно одиночных константных, мостиковых и труднообнаружимых неисправностей в логической схеме.

Задачи:

1. Изучение теоретического материала по теме работы.
2. Ознакомление с принципами работы программного инструмента FSMTest-1.0 построения полного проверяющего теста для конечного автомата.
3. Ознакомление с принципами работы программного инструмента генерации одиночных константных, мостиковых и труднообнаружимых неисправностей в логической схеме.
4. Изучение форматов .bench и .blif описания логических схем, формата .fsm описания конечных автоматов.
5. Построение по формату .bench графического изображения логических схем b02 и s27.
6. Построение по графическому изображению логических схем b02 и s27 соответствующих конечных автоматов.
7. Приведение конечных автоматов для логических схем b02 и s27 к формату .fsm.
8. Построение с помощью программного инструмента FSMTest-1.0 проверяющих тестов для конечных автоматов логических схем b02 и s27.

9. Внесение с помощью программного инструмента генерации неисправностей в логической схеме одиночных константных, мостиковых и труднообнаружимых неисправностей в .blif описания логических схем b02 и s27.

10. Выявление неисправных схем, поведение которых эквивалентно поведению эталонных схем b02 и s27.

11. Экспериментальная оценка полноты проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно неисправных схем, неэквивалентных эталонным схемам b02 и s27.

Методы исследования:


1. Анализ литературы по теме работы.
2. Эксперимент по оценке полноты проверяющего теста.
3. Наблюдение реакций неисправных схем на последовательности проверяющего теста.
4. Сравнение реакций неисправных и эталонной схем на последовательности проверяющего теста.

Организация или отрасль, по тематике которой выполняется работа
техническая диагностика


4 Краткое содержание работы

Рассматриваются одиночные константные, мостиковые и труднообнаружимые неисправности в логической схеме, которые приводят к ошибкам переходов и выходов в соответствующем конечном автомате. Экспериментально оценивается полнота теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно трех указанных типов неисправностей в логической схеме. Эксперименты автоматизируются с помощью программных инструментов генерации проверяющих тестов и одиночных неисправностей в логических схемах.

Руководитель выпускной квалификационной работы
доцент, РФФ

 / С.А. Прокопенко

Задание принял к исполнению
студент 07809 группы

 / Э.Д. Карепин

АННОТАЦИЯ

Выпускная квалификационная работа специалиста состоит из 3 глав, содержит 44 страницы, 31 рисунок, 8 источников, 4 приложения.

ЛОГИЧЕСКАЯ СХЕМА, НЕИСПРАВНОСТЬ, КОНЕЧНЫЙ АВТОМАТ, ПРОВЕРЯЮЩИЙ ТЕСТ, ПОЛНОТА ПРОВЕРЯЮЩЕГО ТЕСТА.

Целью данной работы является экспериментальная оценка полноты проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно одиночных константных, мостиковых и труднообнаружимых неисправностей в логической схеме.

Для проведения экспериментальной оценки полноты проверяющего теста используются две логические схемы b02 и s27. Рассматриваются только одиночные константные, мостиковые и труднообнаружимые неисправности в логической схеме, которые приводят к ошибкам переходов и выходов в соответствующем конечном автомате.

Для каждой рассматриваемой схемы при помощи проверяющего теста, построенного для обнаружения ошибок переходов и выходов в конечном автомате, определяются и убираются из рассмотрения неисправные схемы, поведение которых эквивалентно поведению эталонной схемы. Затем экспериментально оценивается полнота теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно множества неисправных схем, поведение которых отличается от поведения эталонной схемы. Эксперименты автоматизируются с помощью программных инструментов генерации проверяющих тестов и одиночных неисправностей в логических схемах.

ОГЛАВЛЕНИЕ

Введение.....	6
1 Основные определения и обозначения.....	8
1.1 Логические схемы.....	8
1.2 Конечные автоматы.....	9
1.3 Построение полного проверяющего теста для конечного автомата.....	10
1.4 Построение конечного автомата по логической схеме.....	12
2 Построение проверяющих тестов для логических схем b02 и s27.....	15
2.1 Построение конечных автоматов для логических схем b02 и s27.....	15
2.2 Построение проверяющих тестов для конечных автоматов, описывающих поведение логических схем b02 и s27.....	19
3 Экспериментальные результаты по оценке полноты проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно неисправностей в логических схемах.....	28
3.1 Экспериментальные результаты для логической схемы b02.....	29
3.2 Экспериментальные результаты для логической схемы s27.....	34
Заключение.....	37
Список используемой литературы.....	38
Приложение А Формат .fsm описания конечного автомата для логической схемы s27	39
Приложение Б Полный проверяющий тест $T = VW \cup VIW$, построенный для конечного автомата, описывающего поведение логической схемы b02.....	40
Приложение В Полный проверяющий тест $T = VW \cup VIW$, построенный для конечного автомата, описывающего поведение логической схемы s27.....	41
Приложение Г Полный проверяющий тест $T' = VI$, построенный для конечного автомата, описывающего поведение логической схемы s27.....	44

ВВЕДЕНИЕ

Тестирование является одним из важнейших этапов при проектировании, производстве и эксплуатации устройств. Известно, что нельзя построить универсальный проверяющий тест, который был бы способен обнаруживать произвольные неисправности в устройстве. Поэтому, для каждого типа неисправностей строятся свои проверяющие тесты. Однако такие тесты могут обнаруживать и неисправности других типов.

Чтобы иметь возможность строить тесты с гарантированной полнотой, необходимо иметь математическую модель устройства. Поскольку в данной работе мы работаем с логическими схемами [1], то их адекватной математической моделью является конечный автомат [2]. Неисправности в логической схеме приводят как к ошибкам переходов, так и к ошибкам выходов в соответствующем конечном автомате. Если ошибки не приводят к увеличению числа состояний в конечном автомате, то проверяющий тест можно построить методом Василевского [3] следующим образом: $T = VW \cup VIW$, где V – множество достижимости, W – множество различимости, а I – входной алфавит конечного автомата. Такой проверяющий тест хоть и является полным относительно ошибок переходов и выходов, но имеет большую длину.

Известно, что тест $T' = VI$ является полным проверяющим тестом относительно выходных ошибок в конечном автомате. Он имеет меньшую длину, чем тест T , однако, помимо выходных ошибок, может обнаруживать еще и ошибки переходов.

Поэтому в данной работе мы проводим экспериментальную оценку полноты теста T' относительно одиночных неисправностей в логических схемах b02 [4] и s27 [5], которые являются контрольными примерами из сети Интернет.

Работа организована следующим образом. В разделе 1 приведены основные понятия и определения, используемые в работе. Рассказывается, как построить конечный автомат по логической схеме, проведя моделирование поведения схемы на всех входных наборах. Кроме того, вводится понятие модели неисправности и показывается, как построить полный проверяющий тест относительно данной модели неисправности.

В разделе 2 показано, как по логическим схемам b02 и s27 построить соответствующие им конечные автоматы, а затем и проверяющие тесты относительно как ошибок переходов и выходов, так и только ошибок выходов в конечном автомате. Проверяющие тесты строятся с использованием инструмента FSMTest-1.0 [6, 7].

В разделе 3 приведены результаты экспериментов по оценке полноты проверяющего теста $T' = VI$ относительно одиночных константных, мостиковых и труднообнаружимых неисправностей в логических схемах b02 и s27.

В приложении А представлен .fsm формат конечного автомата для логической схемы s27, в приложениях Б и В записаны последовательности проверяющих тестов $T = VW \cup VIW$ для конечных автоматов схем b02 и s27 соответственно, и в приложении Г указаны последовательности проверяющего теста $T' = VI$ для конечного автомата схемы s27.

1 Основные определения и обозначения

В данном разделе рассмотрим основные определения и обозначения, используемые в работе. Рассмотрим, каким образом построить конечный автомат, описывающий поведение логической схемы, а также рассмотрим методы синтеза полных проверяющих тестов для конечных автоматов относительно как ошибок переходов/выходов, так и только ошибок выходов.

1.1 Логические схемы

При проектировании цифровых устройств существуют различные уровни абстракции. Логический уровень является одним из них. На данном уровне устройство представляется в виде совокупности логических элементов [1].

Если поведение устройства не зависит от поданных на него ранее входных воздействий, то для описания устройства можно использовать комбинационную схему (рисунок 1).



Рисунок 1 – Комбинационная схема

Если поведение устройства зависит от поданных на него ранее входных воздействий, то в логической схеме появляются триггеры, которые как раз отвечают за «память» устройства. Такая схема называется последовательностной схемой (рисунок 2).



Рисунок 2 – Последовательностная схема

1.2 Конечные автоматы

Под конечным автоматом [2] понимается пятерка $A = (S, I, O, \psi, \phi)$, где S – конечное множество состояний, I – конечный входной алфавит, O – конечный выходной алфавит, $\psi: S \times I \rightarrow S$ – функция переходов и понимается $\phi: S \times I \rightarrow O$ – функция выходов.

Если в конечном автомате выделено начальное состояние $s_0 \in S$, то такой автомат называется инициальным.

Если в конечном автомате для каждого состояния $s \in S$ определен переход под действием каждого входного символа $i \in I$, то автомат называется полностью определенным, в противном случае частично определенным (или частичным).

Пример 1. Рассмотрим конечный автомат A , таблица переходов-выходов которого изображена на рисунке 3.

$I \backslash S$	s_0	s_1
0	$s_1/0$	$s_1/1$
1	$s_0/1$	$s_0/0$

Рисунок 3 – Конечный автомат A

Множество состояний данного автомата $S = \{s_0, s_1\}$, входной алфавит $I = \{0, 1\}$, выходной алфавит $O = \{0, 1\}$.

Поскольку в каждом состоянии определен переход под действием каждого входного символа, то автомат A является полностью определенным.

Так как в автомате A выделено начальное состояние s_0 , то данный автомат является инициальным.

Рассмотрим автомат $A = (S, I, O, \psi, \phi)$. Состояния s_i и s_j называются эквивалентными, если на любую входную последовательность автомат A в этих состояниях производит одинаковые выходные реакции. Конечный автомат, в котором нет эквивалентных состояний, называется приведенным (минимальным).

Пример 2. Рассмотрим конечный автомат, изображенный на рисунке 4.

$I \backslash S$	s_0	s_1	s_2
0	$s_1/0$	$s_1/1$	$s_1/1$
1	$s_2/1$	$s_1/0$	$s_1/0$

Рисунок 4 – Конечный автомат с эквивалентными состояниями

В данном автомате состояния s_1 и s_2 являются эквивалентными, потому что под действием любого входного символа автомат из этих состояний переходит в одно и то же состояние s_1 и производит одинаковые выходные реакции. Таким образом, данный автомат не является приведенным.

1.3 Построение полного проверяющего теста для конечного автомата

Пусть I^* – множество всех слов в алфавите I , включая и пустое слово ε .

Конечное множество конечных входных последовательностей, по которым в автомате A достижимо каждое состояние из начального состояния, называется множеством достижимости V автомата A .

Конечное множество конечных входных последовательностей, на которые выходные реакции в любых двух состояниях автомата A различны, называется множеством различимости W автомата A .

Пример 3. Рассмотрим конечный автомат A , представленный на рисунке 3.

Его множество достижимости $V = \{\varepsilon, 0\}$: начальное состояние s_0 достигается само из себя по пустой последовательности ε , состояние s_1 достигается из начального состояния s_0 по входному символу 0.

Множество различимости автомата A есть $W = \{0\}$, так как в состоянии s_0 на входной символ 0 автомат производит реакцию 0, а в состоянии s_1 на входной символ 0 – реакцию 1.

Полностью определенный инициальный автомат B называется эквивалентным автомату A (обозначается $B \approx A$), если на любую входную последовательность в начальных состояниях они производят одинаковые реакции.

Пусть A – конечный инициальный приведенный автомат, описывающий поведение эталонной схемы. Пусть неисправности в схеме не увеличивают числа комбинаций состояний триггеров. Тогда \mathfrak{Z} – множество конечных автоматов с тем же входным алфавитом, что и у автомата A и числом состояний не большим, чем у A , которые описывают поведение проверяемых логических схем. Символ \approx обозначает отношение эквивалентности между эталонным автоматом и любым автоматом из множества \mathfrak{Z} . Тройка $\langle A, \approx, \mathfrak{Z} \rangle$ называется моделью неисправности.

Конечное множество конечных входных последовательностей T называется полным проверяющим тестом относительно модели неисправности $\langle A, \approx, \mathfrak{Z} \rangle$, если для всякого автомата из множества \mathfrak{Z} , неэквивалентного эталонному автомату, в множестве T найдется

входная последовательность, на которую реакции эталонного и проверяемого автоматов различны.

Полный проверяющий тест относительно модели неисправности $\langle A, \approx, \mathfrak{Z} \rangle$ можно построить методом Василевского [3]. Множество $T = VW \cup VIW$, где V – множество достижимости, а W – множество различимости эталонного автомата, является полным проверяющим тестом, который обнаруживает все возможные ошибки переходов и выходов в автоматах из множества \mathfrak{Z} .

Метод Василевского состоит из двух этапов. На первом этапе (VW) осуществляется идентификация состояний в проверяемом автомате, то есть мы убеждаемся, что каждому состоянию эталонного автомата A соответствует состояние проверяемого автомата из множества \mathfrak{Z} . На втором этапе (VIW) осуществляется проверка переходов из этих состояний.

Пример 4. Рассмотрим конечный автомат A , представленный на рисунке 3. Для него уже построены множество достижимости $V = \{\varepsilon, 0\}$ и множество различимости $W = \{0\}$. Входной алфавит автомата A есть $I = \{0, 1\}$. Построим полный проверяющий тест для данного автомата методом Василевского: $T = VW \cup VIW$.

Множество VW получается путем приписывания каждой последовательности из множества достижимости V каждой из последовательностей множества различимости W . В данном случае $VW = \{\varepsilon 0, 00\}$.

Множество VIW получается путем приписывания каждого входного символа из множества I к каждой последовательности из множества достижимости V , а затем – приписывания ко всем полученным последовательностям каждой последовательности из множества различимости W . В данном случае $VIW = \{\varepsilon 00, \varepsilon 10, 000, 010\}$.

Проверяющий тест $T = VW \cup VIW = \{\varepsilon 0, 00, \varepsilon 00, \varepsilon 10, 000, 010\}$. После удаления пустого символа ε и поглощаемых последовательностей получим $T = \{10, 000, 010\}$.

Таким образом, полный проверяющий тест для заданного автомата есть $T = \{10, 000, 010\}$. Он состоит из трех входных последовательностей, общая длина теста составляет восемь входных символов.

Если известно, что в автоматах множества \mathfrak{Z} возможны только выходные ошибки, то полный проверяющий тест относительно модели неисправности $\langle A, \approx, \mathfrak{Z} \rangle$ есть $T' = VI$. Данный тест так же может обнаруживать и ошибки переходов, но вопрос полноты в данном случае остается открытым.

Пример 5. Построим проверяющий тест $T' = VI$ для обнаружения выходных ошибок для конечного автомата A , представленного на рисунке 3. Множество достижимости данного автомата $V = \{\varepsilon, 0\}$ и входной алфавит есть $I = \{0, 1\}$.

Проверяющий тест $T' = VI = \{\varepsilon 0, 00, \varepsilon 1, 01\}$. После удаления пустого символа ε и поглощаемых последовательностей получим $T' = \{00, 1, 01\}$.

Таким образом, полный проверяющий тест, построенный для обнаружения выходных ошибок в заданном автомате, есть $T' = \{00, 1, 01\}$. Он состоит из трех входных последовательностей, общая длина теста составляет пять входных символов.

1.4 Построение конечного автомата по логической схеме

Для того чтобы построить конечный автомат $A = (S, I, O, \psi, \phi)$, описывающий поведение логической схемы, необходимо установить соответствие между множествами S, I, O , функциями ψ, ϕ и элементами логической схемы.

Конечное множество S состояний автомата содержит 2^n элементов, где n – количество триггеров в схеме.

Конечное множество I входных символов автомата содержит 2^m элементов, где m – количество входов в схему.

Конечное множество O выходных символов автомата содержит 2^k элементов, где k – количество выходов в схеме.

Функция переходов ψ автомата определяется посредством моделирования поведения схемы на различных входных наборах при различных комбинациях состояний триггеров.

Функция выходов ϕ автомата определяется посредством вычисления значений на выходных полюсах схемы при моделировании поведения схемы на различных входных наборах при различных комбинациях состояний триггеров.

Моделирование поведения логической схемы начинается, когда все триггеры находятся в состоянии 0, то есть начальное состояние конечного автомата соответствует комбинации состояний триггеров $0\dots 0$. При подаче на схему входных воздействий триггеры переходят в новые состояния, то есть достигаются новые комбинации состояний триггеров, а значит, в конечном автомате происходят переходы в новые состояния.

Пример 6. Рассмотрим логическую схему, изображенную на рисунке 6.

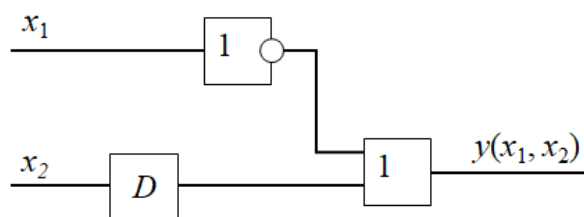


Рисунок 6 –Логическая схема

В данной схеме два входных полюса, один выходной полюс, один элемент памяти (D триггер), один инвертор и один дизъюнктор. Чтобы построить по данной схеме конечный автомат, сначала необходимо поставить в соответствие состояниям, входным и выходным символам автомата состояния входные и выходные наборы схемы.

В конечном автомате два состояния, так как в схеме имеется только один триггер. Состояние 0 конечного автомата соответствует 0 состоянию D триггера, а состояние 1 соответствует состоянию 1 триггера.

Поскольку в схеме имеются два входных полюса, то возможные комбинации входных сигналов схемы суть 00, 01, 10 и 11. Таким образом, в конечном автомате будут четыре входных символа 0, 1, 2 и 3, соответствующие входным наборам схемы.

Поскольку в схеме имеется один выходной полюс, то возможны только два значения сигнала на выходе схемы 0 и 1. Поэтому в конечном автомате будут тоже два выходных символа 0 и 1.

Рассмотрим теперь, как определяются функции переходов и выходов в конечном автомате.

Пусть D триггер находится в состоянии 0 и на вход схемы поступает набор 00. Первый символ 0, пройдя через инвертор, становится 1. Второй символ 0 поступает на D триггер в его состоянии 0. В данном состоянии триггер производит выходную реакцию 0 и остается в состоянии 0. Следовательно на дизъюнктор поступает набор 10, и на выходе схемы получается 1. Таким образом, конечный автомат в состоянии 0 под действием входного символа 0 (что соответствует набору 00 на входе схемы) переходит в состояние 0 и производит выходной символ 1.

Подадим теперь на вход схемы набор 01. Первый символ 0, пройдя через инвертор, становится 1. Второй символ 1 поступает на D триггер в его состоянии 0. В данном состоянии триггер производит выходную реакцию 0 и переходит в состояние 1. Следовательно на дизъюнктор поступает набор 10, и на выходе схемы получается 1. Таким образом, конечный автомат в состоянии 0 под действием входного символа 1 (что

соответствует набору 01 на входе схемы) переходит в состояние 1 и производит выходную реакцию 1.

Промоделировав аналогичным образом поведение схемы на каждом входном наборе, получим конечный автомат, изображенный на рисунке 7.

$L \backslash S$	0	1
0	0/1	0/1
1	1/1	1/1
2	0/0	0/1
3	1/0	1/1

Рисунок 7 – Конечный автомат, построенный по логической схеме на рисунке 6

2 Построение проверяющих тестов для логических схем b02 и s27

Для того чтобы строить качественные проверяющие тесты для логических схем, необходимо иметь адекватную математическую модель. Одной из таких моделей является конечный автомат [2].

В данном разделе мы рассмотрим, каким образом построить конечный автомат по логической схеме, и как построить проверяющий тест для конечного автомата.

2.1 Построение конечных автоматов для логических схем b02 и s27

Для построения конечного автомата по логической схеме можно промоделировать поведение схемы во всех 2^n различных комбинациях триггеров, а потом удалить из автомата недостижимые состояния. Однако мы поступаем другим способом: при моделировании добавляем в автомат новые достижимые состояния, что позволяет ускорить процедуру моделирования, так как не все возможные комбинации состояний триггеров могут быть достижимы.

В данной работе мы использовали две логические схемы b02 [4] и s27 [5]. Обе схемы даны в формате .bench.

На рисунке 8 представлено .bench описание логической схемы b02.

```

# 1 input
# 1 output
# 4 D-type flipflops
# 4 inverters
# 24 gates (1 and, 14 nand, 3 or, 4 not)

INPUT(LINEA)

OUTPUT(U_REG)

U_REG = DFF(U31)
STATO_REG_2_ = DFF(U33)
STATO_REG_1_ = DFF(U38)
STATO_REG_0_ = DFF(U32)

U31 = AND(U35, U37, STATO_REG_2_)
U32 = NAND(U48, U47)
U33 = NAND(U42, U41)
U34 = NOT(STATO_REG_2_)
U35 = NOT(STATO_REG_0_)
U36 = NOT(LINEA)
U37 = NOT(STATO_REG_1_)
U38 = NAND(U52, U51)
U39 = NAND(U50, U49, U35)
U40 = OR(STATO_REG_2_, LINEA)
U41 = NAND(STATO_REG_0_, U40)
U42 = NAND(STATO_REG_1_, U39)
U43 = OR(LINEA, STATO_REG_1_)
U44 = NAND(U34, U43)
U45 = NAND(LINEA, U34)
U46 = NAND(STATO_REG_0_, U45)
U47 = OR(LINEA, STATO_REG_0_, STATO_REG_2_)
U48 = NAND(U46, U37)
U49 = NAND(LINEA, U34)
U50 = NAND(STATO_REG_2_, U36)
U51 = NAND(STATO_REG_0_, U44)
U52 = NAND(STATO_REG_1_, U34, U35)|

```

Рисунок 8 – .bench описание схемы b02

На рисунке 9 представлено .bench описание логической схемы s27.

```

# 4 inputs
# 1 outputs
# 3 D-type flipflops
# 2 inverters
# 8 gates (1 ANDs + 1 NANDs + 2 ORs + 4 NORs)

INPUT(G0)
INPUT(G1)
INPUT(G2)
INPUT(G3)

OUTPUT(G17)

G5 = DFF(G10)
G6 = DFF(G11)
G7 = DFF(G13)

G14 = NOT(G0)
G17 = NOT(G11)

G8 = AND(G14, G6)

G15 = OR(G12, G8)
G16 = OR(G3, G8)

G9 = NAND(G16, G15)

G10 = NOR(G14, G11)
G11 = NOR(G5, G9)
G12 = NOR(G1, G7)
G13 = NOR(G2, G12)

```

Рисунок 9 – .bench описание схемы s27

По данным описаниям мы нарисовали непосредственно сами логические схемы, чтобы дальше было удобно моделировать их поведение и строить конечные автоматы.

На рисунке 10 приведена логическая схема b02.

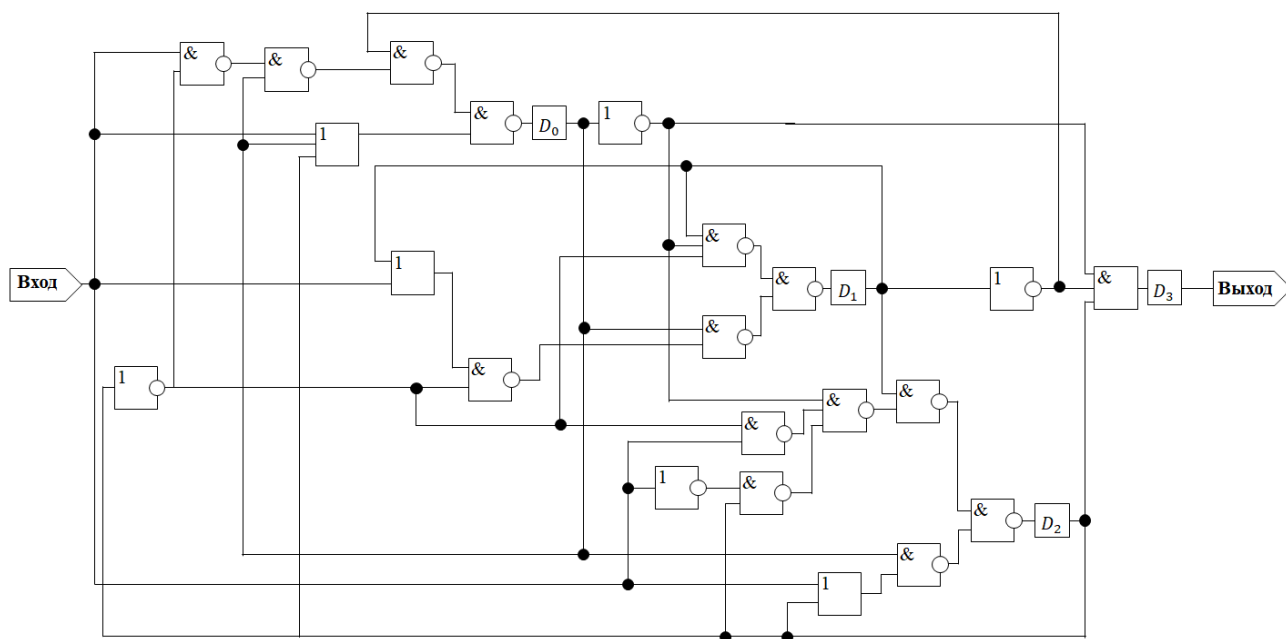


Рисунок 10 – Логическая схема b02

Логическая схема s27 приведена на рисунке 11.

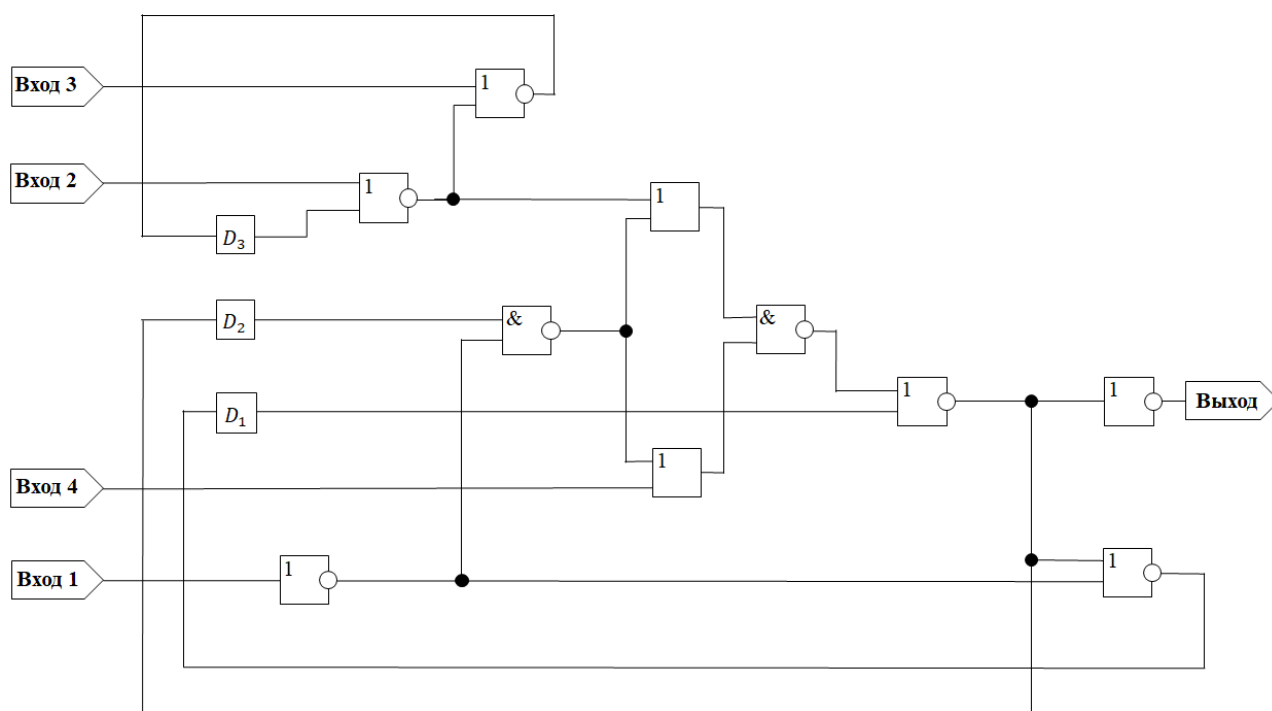


Рисунок 11 – Логическая схема s27

Промоделировав поведение обеих схем на всех входных наборах, мы построили конечные автоматы, описывающие поведение данных схем.

На рисунке 12 изображен конечный автомат, построенный по логической схеме b02.

$\Lambda \backslash S$	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0	$s_1/0$	$s_3/0$	$s_4/0$	$s_5/0$	$s_6/0$	$s_7/0$	$s_7/0$	$s_3/1$
1	$s_1/0$	$s_2/0$	$s_4/0$	$s_4/0$	$s_0/0$	$s_4/0$	$s_7/0$	$s_2/1$

Рисунок 12 – Конечный автомат для схемы b02

В данном автомате два входных символа 0 и 1, поскольку в схеме имеется только один вход; два выходных символа 0 и 1, так как в схеме только один выход; и восемь состояний, хотя в схеме четыре триггера.

Входной символ автомата 0 соответствует значению 0 на входе схемы, входной символ автомата 1 соответствует значению 1 на входе схемы. Выходной символ автомата 0 соответствует значению 0 на выходе схемы, выходной символ автомата 1 соответствует значению 1 на выходе схемы. Состояние s_0 соответствует комбинации 0000 состояний триггеров D_0 , D_1 , D_2 и D_3 , состояние s_1 – комбинации 1000, состояние s_2 – комбинации 1010, состояние s_3 – комбинации 0100, состояние s_4 – комбинации 0110, состояние s_5 – комбинации 1100, состояние s_6 – комбинации 0010 и состояние s_7 – комбинации 1001.

На рисунке 13 изображен конечный автомат, построенный по логической схеме s27.

$\Lambda \backslash S$	s_0	s_1	s_2	s_3	s_4	s_5
0	$s_0/1$	$s_1/1$	$s_2/0$	$s_3/0$	$s_0/1$	$s_1/1$
1	$s_2/0$	$s_1/1$	$s_2/0$	$s_3/0$	$s_0/1$	$s_1/1$
2	$s_0/1$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$	$s_0/1$
3	$s_2/0$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$	$s_0/1$
4	$s_1/1$	$s_1/1$	$s_3/0$	$s_3/0$	$s_1/1$	$s_1/1$
5	$s_1/1$	$s_1/1$	$s_3/0$	$s_3/0$	$s_1/1$	$s_1/1$
6	$s_0/1$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$	$s_0/1$
7	$s_0/1$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$	$s_0/1$
8	$s_4/1$	$s_5/1$	$s_4/1$	$s_5/1$	$s_4/1$	$s_5/1$
9	$s_2/0$	$s_5/1$	$s_2/0$	$s_5/1$	$s_4/1$	$s_5/1$
10	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$
11	$s_2/0$	$s_4/1$	$s_2/0$	$s_4/1$	$s_4/1$	$s_4/1$
12	$s_5/1$	$s_5/1$	$s_5/1$	$s_5/1$	$s_5/1$	$s_5/1$
13	$s_5/1$	$s_5/1$	$s_5/1$	$s_5/1$	$s_5/1$	$s_5/1$
14	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$
15	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$

Рисунок 13 – Конечный автомат для схемы s27

В данном автомате шестнадцать входных символов 0, ..., 15, поскольку в схеме имеются четыре входа; два выходных символа 0 и 1, так как в схеме только один выход; и шесть состояний.

Входной символ автомата 0 соответствует комбинации 0000 на входах схемы Вход 0, Вход 1, Вход 2 и Вход 3, входной символ 1 – комбинации 0001, входной символ 2 – комбинации 0010, входной символ 3 – комбинации 0011, входной символ 4 – комбинации 0100, входной символ 5 – комбинации 0101, входной символ 6 – комбинации 0110, входной символ 7 – комбинации 0111, входной символ 8 – комбинации 1000, входной символ 9 – комбинации 1001, входной символ 10 – комбинации 1010, входной символ 11 – комбинации 1011, входной символ 12 – комбинации 1100, входной символ 13 – комбинации 1101, входной символ 14 – комбинации 1110, входной символ 15 – комбинации 1111.

Выходной символ автомата 0 соответствует значению 0 на выходе схемы, выходной символ автомата 1 соответствует значению 1 на выходе схемы.

Состояние s_0 соответствует комбинации 000 состояний триггеров D_1 , D_2 и D_3 , состояние s_1 – комбинации 001, состояние s_2 – комбинации 010, состояние s_3 – комбинации 011, состояние s_4 – комбинации 100, состояние s_5 – комбинации 101.

2.2 Построение проверяющих тестов для конечных автоматов, описывающих поведение логических схем b02 и s27

При построении проверяющих тестов для логических схем сначала требуется определить, какие неисправности могут возникнуть в схеме. Наиболее распространенными являются одиночные константные, мостиковые (или неисправности «перемычек») и труднообнаружимые неисправности [1, 8].

Константные неисправности возникают только в соединениях между элементами схемы, сами логические элементы функционируют исправно. Константная неисправность фиксирует в заданной линии либо значение сигнала 0, либо значение сигнала 1 вне зависимости от того, какой сигнал в нее подавался. Мостиковые неисправности возникают, если появляется нежелательный проводящий путь между двумя элементами схемы. И, наконец, труднообнаружимые неисправности возникают тогда, когда незначительно меняется поведение логического элемента схемы.

В работе [8] отмечается, что при синтезе тестов для логических схем можно использовать мутационный подход. Для эталонной и проверяемой (возможно неисправной) схем строится входная последовательность, которая отличает проверяемую схему от эталонной (если они не эквивалентны, то есть найдется такая входная последовательность,

по реакции на которую у эталонной и проверяемой схем различны). Проверяющий тест состоит из различающих последовательностей, построенных для множества возможно неисправных схем.

Недостатком такого подхода является то, что как только появляется новая неисправная реализация, тест может оказаться неполным (то есть не обнаруживает неисправности в данной реализации), тогда приходится строить новую различающую последовательность или дорабатывать уже имеющиеся. Кроме того, тесты, построенные по логической схеме, не являются качественными при тестировании, например, СБИС и ПЛИС.

Чтобы строить тесты с гарантированной полнотой, необходимо использовать конечноавтоматные методы синтеза тестов.

В данной работе для построения проверяющих тестов для логических схем b02 и s27 мы использовали инструмент синтеза тестов [6, 7]. Данный инструмент требует, чтобы эталонный конечный автомат был задан в специальном формате .fsm. Поэтому конечные автоматы, построенные для логических схем b02 и s27 и изображенные на рисунках 12 и 13 соответственно, были записаны в формате .fsm.

Конечный автомат для схемы b02 в формате .fsm представлен на рисунке 14.

```
F 0
s 8
i 2
o 2
n0 0
p 16
0 0 1 0
0 1 1 0
1 0 3 0
1 1 2 0
2 0 4 0
2 1 4 0
3 0 5 0
3 1 4 0
4 0 6 0
4 1 0 0
5 0 6 0
5 1 4 0
6 0 7 0
6 1 7 0
7 0 3 1
7 1 2 1
.
```

Рисунок 14 – Конечный автомат для схемы b02 в формате .fsm

В первой строке указывается фиксированная преамбула F 0, такая строка присутствует в .fsm описании любого конечного автомата.

Во второй строке указывается число состояний автомата. В данном случае их 8.

В третьей строке указывается число входных символов автомата. В данном конечном автомате их 2.

В четвертой строке указывается число выходных символов автомата. В данном автомате их 2.

Поскольку автомат инициальный, то в пятой строке указывается начальное состояние автомата p_0 , которое в данном автомате есть 0.

В шестой строке записывается число переходов в автомате. В нашем случае их 16.

В последующих строках записываются сами переходы в формате «начальное_состояние_перехода входной_символ финальное_состояние_перехода выходной_символ». Например, запись 2 0 4 0 означает, что в автомате есть переход из состояния 2 в состояние 4 под действием входного символа 0 с выдачей выходной реакции 0.

Для логической схемы s27 оказалось, что построенный для нее конечный автомат не является приведенным (состояния s_1 и s_5 эквивалентны). Поэтому сначала была проведена минимизация данного конечного автомата и приведенный конечный автомат имеет пять состояний и изображен на рисунке 15.

$I \backslash S$	s_0	s_1	s_2	s_3	s_4
0	$s_0/1$	$s_1/1$	$s_2/0$	$s_3/0$	$s_0/1$
1	$s_2/0$	$s_1/1$	$s_2/0$	$s_3/0$	$s_0/1$
2	$s_0/1$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$
3	$s_2/0$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$
4	$s_1/1$	$s_1/1$	$s_3/0$	$s_3/0$	$s_1/1$
5	$s_1/1$	$s_1/1$	$s_3/0$	$s_3/0$	$s_1/1$
6	$s_0/1$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$
7	$s_0/1$	$s_0/1$	$s_2/0$	$s_2/0$	$s_0/1$
8	$s_4/1$	$s_1/1$	$s_4/1$	$s_1/1$	$s_4/1$
9	$s_2/0$	$s_1/1$	$s_2/0$	$s_1/1$	$s_4/1$
10	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$
11	$s_2/0$	$s_4/1$	$s_2/0$	$s_4/1$	$s_4/1$
12	$s_1/1$	$s_1/1$	$s_1/1$	$s_1/1$	$s_1/1$
13	$s_1/1$	$s_1/1$	$s_1/1$	$s_1/1$	$s_1/1$
14	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$
15	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$	$s_4/1$

Рисунок 15 – Приведенный конечный автомат для схемы s27

Затем данный автомат записан в формате .fsm. Фрагмент описания (18 переходов из 80) представлен на рисунке 16.

```

F 0
s 5
i 16
o 2
n0 0
p 80
0 0 1 0
0 1 1 0
0 2 0 1
0 3 2 0
0 4 1 1
0 5 1 1
0 6 0 1
0 7 0 1
0 8 4 1
0 9 2 0
0 10 4 1
0 11 2 0
0 12 1 1
0 13 1 1
0 14 4 1
0 15 4 1
1 0 1 1
1 1 1 1

```

Рисунок 16 – Фрагмент конечного автомата для схемы s27 в формате .fsm

Полностью формат .fsm конечного автомата для схемы s27 представлен в приложении А.

По .fsm форматам каждого автомата с помощью программного инструмента FSMTest-1.0 (рисунок 17) построен соответствующий проверяющий тест $T = VW \cup VIW$ для обнаружения ошибок переходов и выходов.

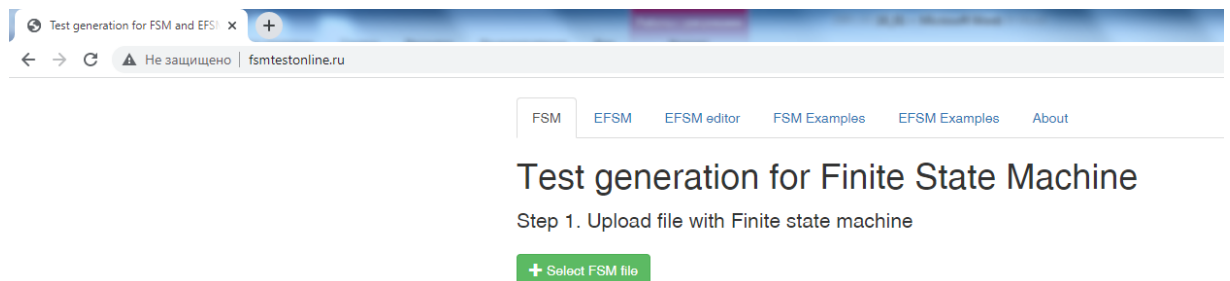


Рисунок 17 – Окно программного инструмента FSMTest-1.0 (Step 1)

Рассмотрим подробно, как работает данный инструмент.

Нажав на кнопку «Select FSM file» в разделе «Step 1. Upload file with Finite State Machine», можно загрузить эталонный конечный автомат в формате .fsm (рисунок 18).

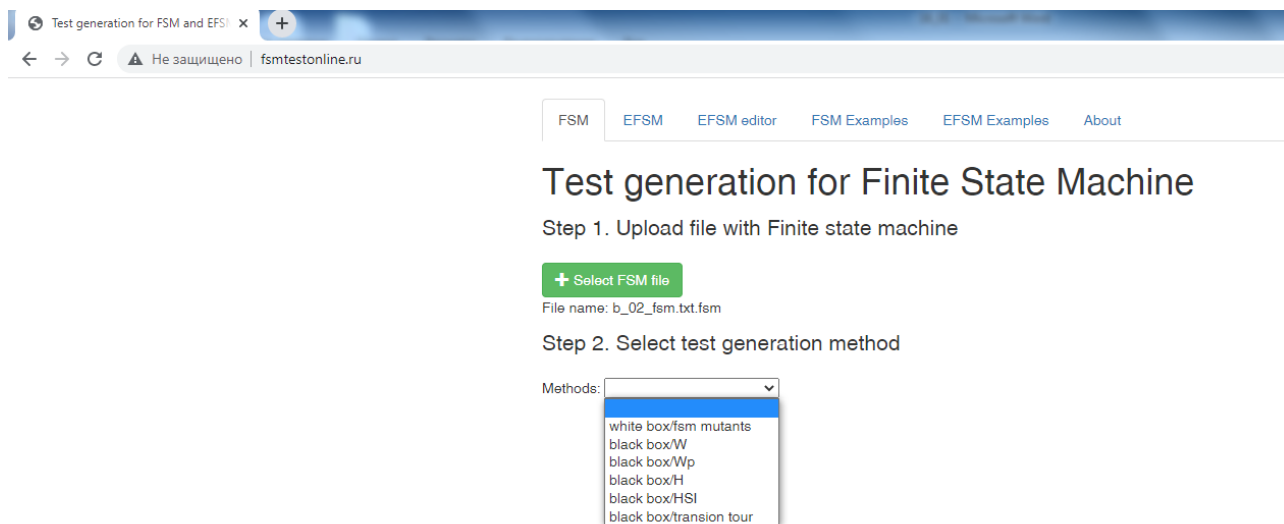


Рисунок 18 – Окно программного инструмента FSMTest-1.0 (Step 2)

После этого появляется раздел «Step 2. Select test generation method», в котором можно выбрать метод синтеза полного проверяющего теста для данного эталонного автомата. Чтобы построить полный проверяющий тест, обнаруживающий все ошибки переходов и выходов, нужно выбрать опцию «black box / W» (рисунок 19).

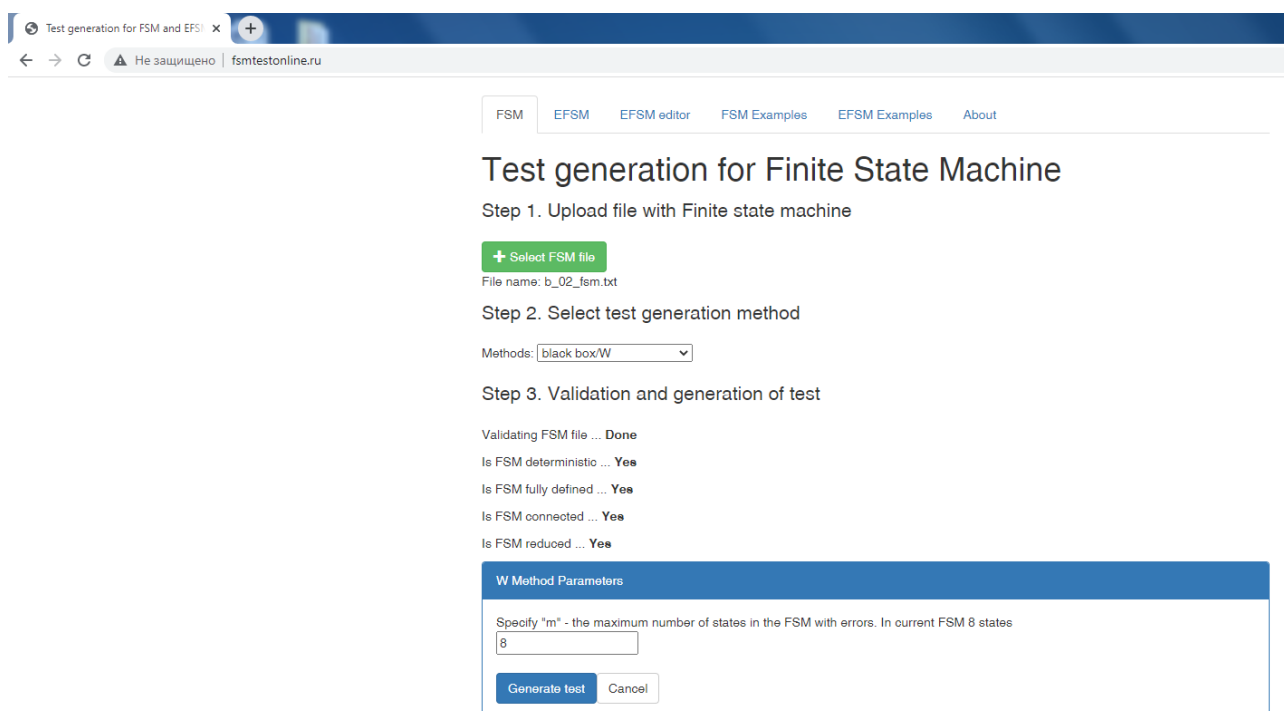


Рисунок 19 – Окно программного инструмента FSMTest-1.0 (Step 3)

Затем появляется раздел «Step 3. Validation and generation of test». Инструмент проверяет, обладает ли эталонный конечный автомат всеми требуемыми для построения

теста характеристиками, и если какое-то требование нарушается, то сигнализирует об этом. Если эталонный автомат удовлетворяет всем требованиям, то в опции «W Method Parameters» можно указать число состояний в проверяемых автоматах. По умолчанию инструмент выставляет число состояний в эталонном автомате.

После нажатия кнопки «Generate test», инструмент генерирует проверяющий тест выбранным методом и выдает его последовательности в формате «входной символ_1/выходной символ_1 входной символ_2/выходной символ_2 ... входной символ_п/выходной символ_п».

Мы полагаем, что неисправности не увеличивают числа состояний в автомате. На рисунке 20 представлен фрагмент окна программного инструмента FSMTest-1.0, в котором приведена часть последовательностей проверяющего теста для конечного автомата логической схемы b02, построенного методом Василевского в классе автоматов с числом состояний не более чем у эталонного автомата.

FSM

EFSM

EFSM editor

FSM Examples

EFSM Examples

About

Test generation for Finite State Machine

Step 1. Upload file with Finite State Machine

+ Select FSM file

File name: b_02_fsm.txt

Step 2. Select test generation method

Methods: black box/W

Step 3. Validation and generation of test

Validating FSM file ... Done

Is FSM deterministic ... Yes

Is FSM fully defined ... Yes

Is FSM connected ... Yes

Is FSM reduced ... Yes

W Method Parameters

Specify "m" - the maximum number of states in the FSM with errors. In current FSM 8 states

8

Generate test

Cancel

Generating test sequences ... Done

Test generation time: 0.047 sec

TEST SEQUENCES

0/0 0/0 0/0 0/0 0/0 0/1 0/0 0/0
0/0 0/0 0/0 0/0 0/0 0/1 0/0 1/0 0/0
0/0 0/0 0/0 0/0 0/0 0/1 1/0 0/0
0/0 0/0 0/0 0/0 0/0 1/1 0/0 0/0 0/0
0/0 0/0 0/0 0/0 1/0 0/1 0/0 0/0
0/0 0/0 0/0 1/0 0/0 0/0 0/1 0/0
0/0 0/0 0/0 1/0 0/0 0/0 0/1 1/0 0/0
0/0 0/0 0/0 1/0 0/0 0/0 1/1 0/0
0/0 0/0 0/0 1/0 0/0 1/0 0/1 0/0 0/0
0/0 0/0 0/0 1/0 1/0 0/0 0/0 0/0

Рисунок 20 – Окно программного инструмента FSMTest-1.0 с фрагментом проверяющего теста $T = VW \cup VIW$ для схемы b02

Полный проверяющий тест $T = VW \cup VIW$ содержит 45 входных последовательностей, общая длина теста составляет 378 входных символов. В приложении Б представлены все последовательности данного проверяющего теста.

Аналогичным образом строится полный проверяющий тест $T' = VI$ для обнаружения только выходных ошибок в конечном автомате. Такой тест для конечного автомата схемы b02 представлен на рисунке 21 и содержит 9 входных последовательностей. Общая длина теста составляет 37 входных символов.

```

0/0 1/0
1/0 1/0 0/0
0/0 0/0 1/0
1/0 1/0 1/0 0/0
1/0 1/0 1/0 1/0
0/0 0/0 0/0 1/0
0/0 0/0 0/0 0/0 1/0
0/0 0/0 0/0 0/0 0/1
0/0 0/0 0/0 0/0 1/1

```

Рисунок 21 –Проверяющий тест для обнаружения выходных ошибок в конечном автомате схемы b02

Полный проверяющий тест $T = VW \cup VIW$ для конечного автомата схемы s27 содержит 228 входных последовательностей, общая длина теста составляет 769 входных символов. Фрагмент теста представлен на рисунке 22.

The screenshot shows the FSMTest-1.0 web interface. It has a navigation bar with links: FSM, EFSM, EFSM editor, FSM Examples, EFSM Examples, and About. The main heading is "Test generation for Finite State Machine".

Step 1. Upload file with Finite State Machine
 A green button labeled "+ Select FSM file" is shown. Below it, the file name is "s_27_fsm.txt_min.txt".

Step 2. Select test generation method
 A dropdown menu labeled "Methods:" shows "black box/W" selected.

Step 3. Validation and generation of test
 A series of status checks are shown: "Validating FSM file ... Done", "Is FSM deterministic ... Yes", "Is FSM fully defined ... Yes", "Is FSM connected ... Yes", and "Is FSM reduced ... Yes".

W Method Parameters
 A blue header box contains the text "Specify 'm' - the maximum number of states in the FSM with errors. In current FSM 5 states". Below this is a text input field containing the number "5". At the bottom of this box are two buttons: "Generate test" and "Cancel".

Below the parameters box, the status "Generating test sequences ... Done" and "Test generation time: 0.047 sec Done" are displayed.

TEST SEQUENCES
 A scrollable area shows a list of test sequences, each consisting of input/output pairs separated by slashes. The visible sequences are:
 0/0 0/1 0/1 0/1
 0/0 0/1 2/1
 0/0 0/1 9/1
 0/0 1/1 0/1 0/1
 0/0 1/1 2/1
 0/0 1/1 9/1
 0/0 2/1 0/0 0/1
 0/0 2/1 2/1
 0/0 2/1 9/0

Рисунок 22 – Окно программного инструмента FSMTest-1.0 с фрагментом проверяющего теста $T = VW \cup VIW$ для схемы s27

В приложении В представлены все последовательности данного проверяющего теста.

На рисунке 23 представлен фрагмент теста $T' = VI$ для обнаружения ошибок выходов в конечном автомате для схемы s27. Тест содержит 76 входных последовательностей, общая длина теста составляет 170 входных символов.

0/1
 1/0
 2/0
 5/1
 6/1
 7/1
 8/1
 9/0
 10/1
 11/0
 12/1
 13/1
 14/1
 15/1
 4/1 0/1
 4/1 1/1
 4/1 2/1
 4/1 3/1
 4/1 4/1
 4/1 5/1
 4/1 6/1
 4/1 7/1
 4/1 8/1
 4/1 9/1
 4/1 11/1
 4/1 12/1
 4/1 13/1
 4/1 14/1
 4/1 15/1
 3/0 0/0

Рисунок 23 – Фрагмент проверяющего теста для обнаружения выходных ошибок в конечном автомате для схемы s27

Все последовательности данного проверяющего теста представлены в приложении Г.

3 Экспериментальные результаты по оценке полноты проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате, относительно неисправностей в логических схемах

В данной работе мы рассматриваем в логических схемах только одиночные неисправности трех типов: константные, мостиковые и труднообнаружимые. Эти неисправности в схеме приводят к ошибкам переходов и выходов в соответствующем конечном автомате. Для проведения экспериментов необходимо выполнить следующие действия:

- а) внести в логическую схему одиночные неисправности;
- б) с помощью построенного проверяющего теста $T = VW \cup VIW$ выявить те схемы, поведение которых эквивалентно поведению эталонной схемы, и далее убрать их из рассмотрения;
- в) подать на оставшиеся схемы проверяющий тест $T' = VI$ и оценить его полноту.

Эксперименты проводились с помощью программного инструмента, используемого [8].

На вход инструмента подаются:

- а) логическая схема в формате .blif;
- б) проверяющий тест $T = VW \cup VIW$, построенный для конечного автомата, описывающего поведение эталонной схемы;
- в) соответствие между входными символами автомата и входными сигналами логической схемы.

В результате работы программного инструмента по эталонной логической схеме, заданной в .blif описании, генерируются три множества схем: одно с одиночными константными неисправностями, второе – с одиночными мостиковыми неисправностями и третье – с одиночными труднообнаружимыми неисправностями. Следует отметить, что данный программный инструмент вносит неисправность в любой элемент логической схемы случайным образом.

Затем на каждую неисправную схему и на эталонную схему подается проверяющий тест $T = VW \cup VIW$, построенный по конечному автомату эталонной схемы. Именно для этого программному инструменту и требуется строгое соответствие между входными символами автомата и входными сигналами логической схемы. В результате программный инструмент выдает список схем, которые не обнаруживаются проверяющим тестом, то есть

их поведение эквивалентно поведению эталонной логической схемы. Такие схемы мы далее не рассматриваем.

Теперь на оставшиеся неисправные схемы подается проверяющий тест $T' = VI$. В результате программный инструмент выдает список тех схем, которые не обнаруживаются данным тестом. Поскольку тест $T' = VI$ способен обнаруживать все схемы, неисправности в которых приводят к выходным ошибкам в соответствующем конечном автомате, и, возможно, некоторые ошибки переходов, то в полученном списке будут схемы, неисправности в которых приводят только к ошибкам переходов в конечном автомате. Для каждого типа неисправностей в логической схеме программный инструмент считает процент обнаруженных схем.

3.1 Экспериментальные результаты для логической схемы b02

По .blif описанию логической схемы b02 с использованием программного инструмента сгенерированы три множества неисправных схем (рисунок 24).

```
user01@user01-VirtualBox: ~/benchmark_testing
user01@user01-VirtualBox:~/benchmark_testing$ python3 ./scripts/main.py b02
SSF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/b02_circuit_test.txt_MatrixCovSSF
RESULTS:
Total number of mutants: 48
Coverage: 100.0%
List of unkillable SSF mutants:
-----
1
SIF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/b02_circuit_test.txt_MatrixCovSIF
RESULTS:
Total number of mutants: 23
Coverage: 95.7%
List of unkillable SIF mutants:
-----
2
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U43_LINEA_STATO_REG_2_.blif
-----
SGF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/b02_circuit_test.txt_MatrixCovSGF
RESULTS:
Total number of mutants: 23
Coverage: 78.3%
List of unkillable SGF mutants:
-----
3
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U32_U47_1_-_1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U40_LINEA_1_-_0.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U42_U39_1_-_1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U47_LINEA_2_-_0.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U51_U44_1_-_1.blif
-----
user01@user01-VirtualBox:~/benchmark_testing$
```

Рисунок 24 – Окно программного инструмента с результатами обнаружения неисправностей проверяющим тестом $T = VW \cup VIW$ для схемы b02

В области, пронумерованной цифрой 1 на рисунке 24, содержится информация об одиночных константных неисправностях (SSF faults), внесенных в эталонную логическую схему b02. Число схем с таким типом неисправности (строка «Total number of mutants») равно 48. С целью выявления неисправностей, не портящих внешнего поведения схемы, на неисправные схемы подавался тест $T = VW \cup VIW$. Все одиночные константные неисправности в схемах были обнаружены данным тестом (строка «Coverage: 100.0 %»).

В области, пронумерованной на рисунке 24 цифрой 2, содержится информация об одиночных мостиковых неисправностях (SIF faults), внесенных в эталонную логическую схему. Число схем с таким типом неисправности (строка «Total number of mutants») равно 23. Проверяющий тест $T = VW \cup VIW$ показал, что одна из мостиковых неисправностей оказалась необнаружимой (строка «Coverage: 95.7 %» и в списке схем с необнаружимыми неисправностями (List of unkilld SIF mutants) указан путь к файлу b02_U43_LINEA_STATO_REG_2_.blif с неисправной схемой).

Для того чтобы посмотреть, какова же эта мостиковая неисправность, мы нарисовали неисправную схему (рисунок 25).

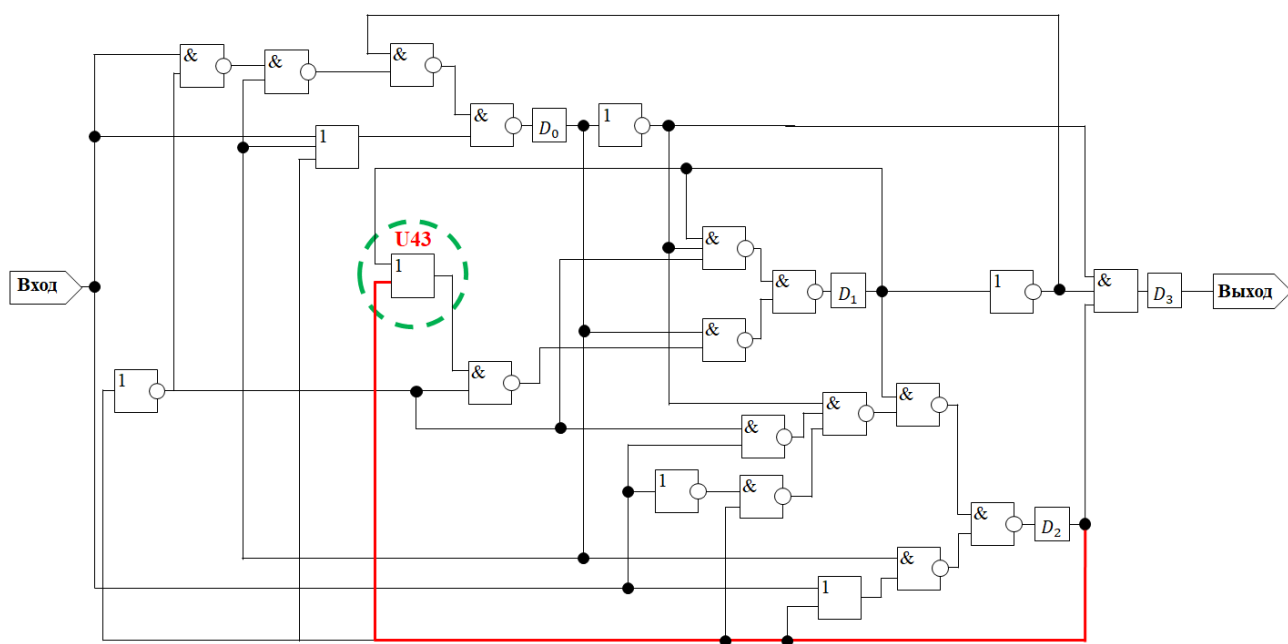


Рисунок 25 – Схема b02 с необнаружимой мостиковой неисправностью

В данной схеме один из входов дизъюнктора (элемент U43 в .blif описании, выделен пунктирной линией) изменился: со входного полюса (как видно на рисунке 10) замкнулся на выход триггера D_2 . Однако данная ошибка не меняет внешнего поведения схемы, то есть схема с неисправностью эквивалентна эталонной схеме.

В области, пронумерованной на рисунке 24 цифрой 3, содержится информация об одиночных труднообнаружимых неисправностях (SGF faults), внесенных в эталонную логическую схему. Число схем с таким типом неисправности (строка «Total number of mutants») равно 23. Проверяющий тест $T = VW \cup VIW$ показал, что пять неисправностей не обнаруживаются (строка «Coverage: 78.3 %» и в списке схем с необнаружимыми неисправностями (List of unkilld SIF mutants) указаны пути к файлам в формате .blif с описаниями неисправных схем).

На рисунке 26 пунктирными линиями выделены неисправные элементы.

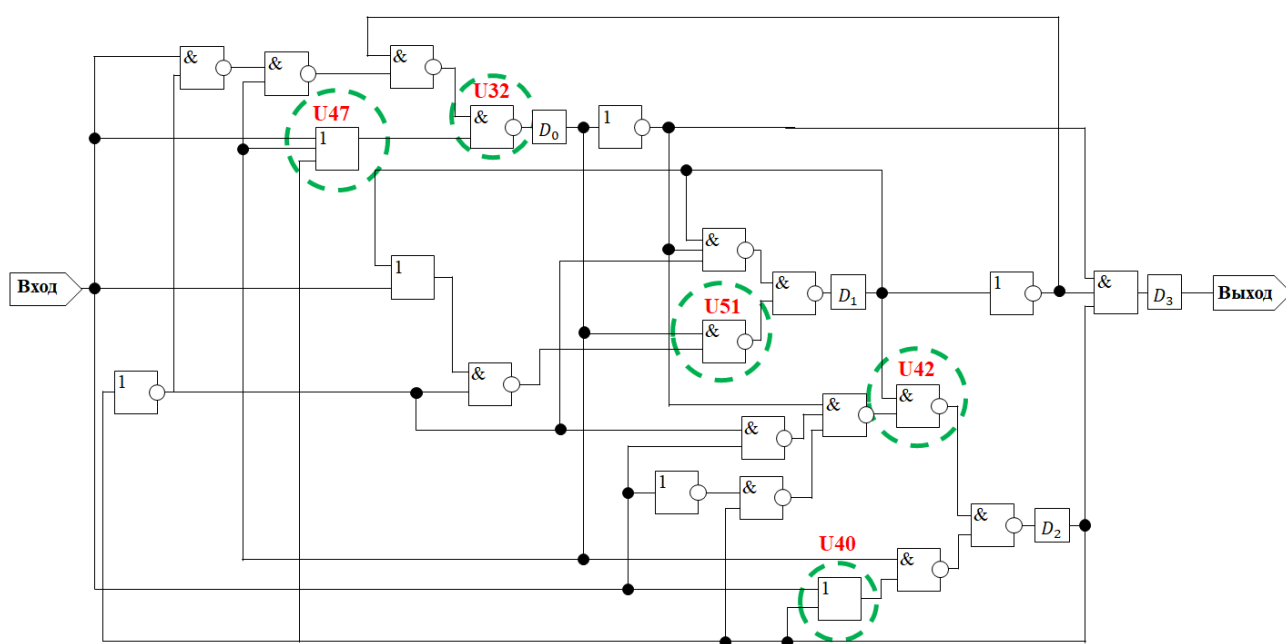


Рисунок 26 – Схема b02 с неисправными элементами

Элементы, занумерованные в .blif описании логической схемы b02 U47 и U40, являются дизъюнкторами. Элементы, занумерованные U32, U51 и U42, являются элементами И-НЕ, реализуемая ими функция есть штрих Шеффера (отрицание конъюнкции).

Мы проанализировали данные пять необнаружимых неисправностей. Труднообнаружимые неисправности должны изменять функцию логического элемента в схеме. Поскольку неисправности вносятся в .blif описание схемы, то оказалось, что все пять неисправностей не меняют саму функцию элемента, а меняют только представление этой функции в виде дизъюнктивной нормальной формы (используемый программный инструмент расценивает такие изменения как труднообнаружимые неисправности).

На рисунке 27 приведен фрагмент файла b02.blif с описанием функции логического элемента U40 эталонной схемы.

```

33 .names STAT0_REG_2_ LINEA U40
34 1- 1
35 -1 1

```

Рисунок 27 – Описание эталонного поведения элемента U40

Согласно данному описанию U40 является дизъюнктом и его ДНФ есть $x_1 \vee x_2$. После внесения программным инструментом труднообнаружимой неисправности в описание данного элемента (рисунок 28), получили, что U40 по-прежнему является дизъюнктом, но его ДНФ есть $x_1 \overline{x_2} \vee x_2$.

```

33 .names STAT0_REG_2_ LINEA U40
34 10 1
35 -1 1

```

Рисунок 28 – Описание поведения элемента U40 в файле b02_U40_LINEA_1_-_0.blif

Такая неисправность хоть и не является настоящей физической неисправностью, но может привести в дальнейшем к неисправностям, если реализовывать полученную формулу, например, на программируемой логической матрице.

Таким образом, из 94 неисправных схем только 88 схем имеют неисправности, обнаружимые проверяющим тестом $T = VW \cup VIW$, то есть эти схемы не эквивалентны эталонной схеме b02.

Теперь мы хотим оценить, сколько неисправных схем из этих 88 будут обнаружены проверяющим тестом $T' = VI$, построенным для обнаружения выходных ошибок в конечном автомате.

На рисунке 29 представлен результат работы программного инструмента, в котором видно, какие неисправные схемы не обнаруживаются тестом $T' = VI$. Другими словами, оценивается полнота теста $T' = VI$ относительно обнаружимых одиночных константных, мостиковых и труднообнаружимых неисправностей в логической схеме b02.


```

user01@user01-VirtualBox:~/benchmark_testing$ python3 ./scripts/main.py b02
SSF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/b02_circuit_test.txt_MatrixCovSSF

RESULTS:
Total number of mutants: 48
Coverage: 77.1%
List of unkilld SSF mutants:
-----
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_LINEA_S0.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U34_S1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U36_S0.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U36_S1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U37_S1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U40_S0.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U41_S1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U46_S0.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U48_S1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U49_S1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SSFMutants/b02_U50_S1.blif
-----

SIF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/b02_circuit_test.txt_MatrixCovSIF

RESULTS:
Total number of mutants: 22
Coverage: 68.2%
List of unkilld SIF mutants:
-----
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U34_STATO_REG_2_U_REG.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U36_LINEA_U_REG.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U39_U49_STATO_REG_1_.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U41_STATO_REG_0_STATO_REG_1_.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U48_U37_LINEA.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U49_LINEA_U_REG.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SIFMutants/b02_U50_STATO_REG_2__U_REG.blif
-----

SGF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/b02_circuit_test.txt_MatrixCovSGF

RESULTS:
Total number of mutants: 18
Coverage: 72.2%
List of unkilld SGF mutants:
-----
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U36_LINEA_1_0_1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U37_STATO_REG_1_1_0_.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U39_U49_1_-0_.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U41_STATO_REG_0_1_0_1.blif
/home/user01/benchmark_testing//benchmarks/b02_mutants/SGFMutants/b02_U46_U45_1_-0_.blif
-----

user01@user01-VirtualBox:~/benchmark_testing$

```

Рисунок 29 – Окно программного инструмента с результатами оценки полноты теста $T' = VI$ для схемы b02

Экспериментальные результаты показали, что полнота данного теста относительно одиночных константных неисправностей составляет 77,1 % (не обнаруживаются 11 неисправных схем из 48; область, отмеченная цифрой 1 на рисунке 29); относительно одиночных мостиковых неисправностей – 68,2 % (не обнаруживаются 7 неисправных схем из 22; область, отмеченная цифрой 2 на рисунке 29); относительно одиночных труднообнаружимых неисправностей – 72,2 % (не обнаруживаются 5 неисправных схем из 18; область, отмеченная цифрой 3 на рисунке 29). В среднем полнота теста $T' = VI$ составляет 73,9 % относительно множества схем, неисправности в которых приводят как к ошибкам переходов, так и к ошибкам выходов в соответствующих конечных автоматах.

3.2 Экспериментальные результаты для логической схемы s27

Для логической схемы s27 сгенерированы три множества неисправных схем: 28 с одиночными константными неисправностями, 10 с одиночными мостиковыми неисправностями и 10 с одиночными труднообнаружимыми неисправностями. Затем на эти схемы подавался тест $T = VW \cup VIW$. Все неисправности в схемах были обнаружены данным тестом (рисунок 30).

```
user01@user01-VirtualBox:~/benchmark_testing$ python3 ./scripts/main.py s27
SSF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/s27_circuit_test.txt_MatrixCovSSF

RESULTS:
Total number of mutants: 28
Coverage: 100.0%
List of unkilld SSF mutants:
-----
-----

SIF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/s27_circuit_test.txt_MatrixCovSIF

RESULTS:
Total number of mutants: 10
Coverage: 100.0%
List of unkilld SIF mutants:
-----
-----

SGF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/s27_circuit_test.txt_MatrixCovSGF

RESULTS:
Total number of mutants: 10
Coverage: 100.0%
List of unkilld SGF mutants:
-----
-----

user01@user01-VirtualBox:~/benchmark_testing$
```

Рисунок 30 – Окно программного инструмента с результатами обнаружения неисправностей проверяющим тестом $T = VW \cup VIW$ для схемы s27

В области, пронумерованной цифрой 1 на рисунке 30, содержится информация об одиночных константных неисправностях (SSF faults), внесенных в эталонную логическую схему s27. Число схем с таким типом неисправности (строка «Total number of mutants») равно 28. С целью выявления неисправностей, не портящих внешнего поведения схемы, на неисправные схемы подавался тест $T = VW \cup VIW$. Все одиночные константные неисправности в схемах были обнаружены данным тестом (строка «Coverage: 100.0 %»).

В области, пронумерованной на рисунке 30 цифрой 2, содержится информация об одиночных мостиковых неисправностях (SIF faults), внесенных в эталонную логическую схему. Число схем с таким типом неисправности (строка «Total number of mutants») равно 10.

Проверяющий тест $T = VW \cup VIW$ показал, что все мостиковые неисправности обнаруживаются данным тестом (строка «Coverage: 100.0 %»).

В области, пронумерованной на рисунке 30 цифрой 3, содержится информация об одиночных труднообнаружимых неисправностях (SGF faults), внесенных в эталонную логическую схему. Число схем с таким типом неисправности (строка «Total number of mutants») равно 10. Проверяющий тест $T = VW \cup VIW$ показал, что все неисправности обнаруживаются (строка «Coverage: 100.0 %»).

Таким образом, получили 48 схем, неисправности в которых обнаруживаются проверяющим тестом $T = VW \cup VIW$, то есть эти схемы не эквивалентны эталонной схеме s27.

Теперь мы хотим оценить, сколько неисправных схем из этих 48 будут обнаружены проверяющим тестом $T' = VI$ для выходных ошибок.

На рисунке 31 представлено окно программного инструмента, в котором видно, какие неисправные схемы не обнаруживаются тестом $T' = VI$, то есть оценена полнота теста $T' = VI$ относительно обнаружимых одиночных константных, мостиковых и труднообнаружимых неисправностей в логической схеме s27.

```
user01@user01-VirtualBox:~/benchmark_testing$ python3 ./scripts/main.py s27
SSF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/s27_circuit_test.txt_MatrixCovSSF

RESULTS:
Total number of mutants: 28
Coverage: 96.4%
List of unkilld SSF mutants:
-----
/home/user01/benchmark_testing//benchmarks/s27_mutants/SSFMutants/s27_G2_S0.blif
-----

SIF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/s27_circuit_test.txt_MatrixCovSIF

RESULTS:
Total number of mutants: 10
Coverage: 90.0%
List of unkilld SIF mutants:
-----
/home/user01/benchmark_testing//benchmarks/s27_mutants/SIFMutants/s27_n22_new_G12_G5.blif
-----

SGF faults
Deriving matrix of coverage: /home/user01/benchmark_testing//benchmarks/s27_circuit_test.txt_MatrixCovSGF

RESULTS:
Total number of mutants: 10
Coverage: 100.0%
List of unkilld SGF mutants:
-----
-----

user01@user01-VirtualBox:~/benchmark_testing$
```

Рисунок 31 – Окно программного инструмента с результатами оценки полноты теста $T = VI$ для схемы s27

Экспериментальные результаты показали, что полнота теста $T' = VI$ относительно одиночных константных неисправностей составляет 96,4 % (не обнаруживается 1 неисправная схема из 28; отмечено цифрой 1 на рисунке 31); относительно одиночных мостиковых неисправностей – 90 % (не обнаруживается 1 неисправная схема из 10; отмечено цифрой 2 на рисунке 31); относительно одиночных труднообнаружимых неисправностей – 100 % (отмечено цифрой 3 на рисунке 31). В среднем полнота теста $T' = VI$ составляет 95,8 % относительно множества схем, неисправности в которых приводят как к ошибкам переходов, так и к ошибкам выходов в соответствующих конечных автоматах.

ЗАКЛЮЧЕНИЕ

В работе проведена экспериментальная оценка полноты проверяющего теста, построенного для обнаружения выходных ошибок в конечном автомате, описывающем поведение логической схемы, относительно одиночных константных, мостиковых и труднообнаружимых неисправностей в схеме. Для проведения экспериментов использовались логические схемы b02 и s27.

Изучены .bench и .blif описания схем. По .bench описаниям логических схем b02 и s27 графически изображены соответствующие схемы, промоделировано их поведение и построены соответствующие им конечные автоматы.

Изучены принципы работы программного инструмента FSMTest-1.0 построения полного проверяющего теста для конечного автомата. Для конечных автоматов схем b02 и s27 с помощью программного инструмента FSMTest-1.0 построены проверяющие тесты $T = VW \cup VIW$ и $T' = VI$. С этой целью изучен формат .fsm описания конечных автоматов, и конечные автоматы для схем b02 и s27 записаны в данном формате.

Изучены принципы работы программного инструмента генерации неисправностей в логической схеме. С помощью данного программного инструмента в .blif описание каждой схемы внесены одиночные константные, мостиковые и труднообнаружимые неисправности и проведены эксперименты по оценке полноты проверяющего теста $T' = VI$ относительно внесенных в схему неисправностей.

Экспериментальные результаты показали, что полнота теста $T' = VI$, построенного для обнаружения выходных ошибок в конечном автомате, является достаточно высокой при обнаружении ошибок переходов/выходов, к которым приводят одиночные неисправности в логической схеме. В среднем полнота теста $T' = VI$ относительно одиночных константных, мостиковых и труднообнаружимых неисправностей в логической схеме b02 составляет более 73,9 %, а в схеме s27 – 95,8 %.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Скобцов Ю.А., Скобцов В.Ю. Логическое моделирование и тестирование цифровых устройств. Донецк: ИПММ НАН Украины, ДонНТУ, 2005 – 436 с.
2. Гилл, А. Введение в теорию конечных автоматов / А. Гилл М.: Наука, 1966. – 272 с.
3. Василевский, М.П. О распознавании неисправностей автоматов. / М.П. Василевский. // Кибернетика, 9(4) – 1973. – С. 93-108.
4. PoliTo ITC99 (I99T) – URL: <https://github.com/cad-polito-it/I99T> (дата обращения 8.11.2023).
5. ISCAS89 Sequential Benchmark Circuits – URL: <https://filebox.ece.vt.edu/~mhsiao/iscas89.html> (дата обращения 8.11.2023).
6. Test generation for Finite State Machine – URL: <http://www.fsmtestonline.ru> (дата обращения 8.11.2023).
7. Shabaldina N., Gromov M. FSMTest-1.0: a manual for researches // Proc. of IEEE East-West Design & Test Symposium (EWDTS'2015). Batumi. Georgia. September 26–29, 2015.– P. 216–219.
8. Лапутенко А. В. Синтез тестов для цифровых систем на высоком и низком уровнях абстракции / А. В. Лапутенко, Е. М. Винарский // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2019. № 47. С. 110-117. URL: <http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000661796>

ПРИЛОЖЕНИЕ А

Формат .fsm описания конечного автомата для логической схемы s27

F 0	2 5 3 0
s 5	2 6 2 0
i 16	2 7 2 0
o 2	2 8 4 1
n0 0	2 9 2 0
p 80	2 10 4 1
0 0 1 0	2 11 2 0
0 1 1 0	2 12 1 1
0 2 0 1	2 13 1 1
0 3 2 0	2 14 4 1
0 4 1 1	2 15 4 1
0 5 1 1	3 0 3 0
0 6 0 1	3 1 3 0
0 7 0 1	3 2 2 0
0 8 4 1	3 3 2 0
0 9 2 0	3 4 3 0
0 10 4 1	3 5 3 0
0 11 2 0	3 6 2 0
0 12 1 1	3 7 2 0
0 13 1 1	3 8 1 1
0 14 4 1	3 9 1 1
0 15 4 1	3 10 4 1
1 0 1 1	3 11 4 1
1 1 1 1	3 12 1 1
1 2 0 1	3 13 1 1
1 3 0 1	3 14 4 1
1 4 1 1	3 15 4 1
1 5 1 1	4 0 0 1
1 6 0 1	4 1 0 1
1 7 0 1	4 2 0 1
1 8 1 1	4 3 0 1
1 9 1 1	4 4 1 1
1 10 4 1	4 5 1 1
1 11 4 1	4 6 0 1
1 12 1 1	4 7 0 1
1 13 1 1	4 8 4 1
1 14 4 1	4 9 4 1
1 15 4 1	4 10 4 1
2 0 2 0	4 11 4 1
2 1 2 0	4 12 1 1
2 2 2 0	4 13 1 1
2 3 2 0	4 14 4 1
2 4 3 0	4 15 4 1

ПРИЛОЖЕНИЕ Б

Полный проверяющий тест $T = VW \cup VIW$, построенный для конечного автомата,
описывающего поведение логической схемы b02

- 1) 0/0 0/0 0/0 0/0 0/0 0/1 0/0 0/0
- 2) 0/0 0/0 0/0 0/0 0/0 0/1 0/0 1/0 0/0
- 3) 0/0 0/0 0/0 0/0 0/0 0/1 1/0 0/0
- 4) 0/0 0/0 0/0 0/0 0/0 1/1 0/0 0/0 0/0
- 5) 0/0 0/0 0/0 0/0 1/0 0/1 0/0 0/0
- 6) 0/0 0/0 0/0 1/0 0/0 0/0 0/1 0/0
- 7) 0/0 0/0 0/0 1/0 0/0 0/0 0/1 1/0 0/0
- 8) 0/0 0/0 0/0 1/0 0/0 0/0 1/1 0/0
- 9) 0/0 0/0 0/0 1/0 0/0 1/0 0/1 0/0 0/0
- 10) 0/0 0/0 0/0 1/0 1/0 0/0 0/0 0/0
- 11) 0/0 0/0 1/0 0/0 0/0 0/1 0/0
- 12) 0/0 0/0 1/0 0/0 0/0 0/1 1/0 0/0
- 13) 0/0 0/0 1/0 0/0 0/0 1/1 0/0
- 14) 0/0 0/0 1/0 0/0 1/0 0/1 0/0 0/0
- 15) 0/0 0/0 1/0 1/0 0/0 0/0 0/0
- 16) 0/0 1/0 0/0 0/0 0/0 0/1 0/0 0/0 0/0 0/1
- 17) 0/0 1/0 0/0 0/0 0/0 0/1 0/0 0/0 0/0 1/1 0/0
- 18) 0/0 1/0 0/0 0/0 0/0 0/1 0/0 0/0 1/0 0/1
- 19) 0/0 1/0 0/0 0/0 0/0 0/1 0/0 1/0 0/0 0/0 0/1
- 20) 0/0 1/0 0/0 0/0 0/0 0/1 1/0 0/0 0/0 0/1
- 21) 0/0 1/0 0/0 0/0 0/0 1/1 0/0 0/0 0/0 0/1
- 22) 0/0 1/0 0/0 0/0 0/0 1/1 0/0 0/0 0/0 1/1 0/0
- 23) 0/0 1/0 0/0 0/0 0/0 1/1 0/0 0/0 1/0 0/1
- 24) 0/0 1/0 0/0 0/0 0/0 1/1 0/0 1/0 0/0 0/0 0/0
- 25) 0/0 1/0 0/0 0/0 0/0 1/1 1/0 0/0 0/0 0/1
- 26) 0/0 1/0 0/0 0/0 1/0 0/1 0/0 0/0 0/0
- 27) 0/0 1/0 0/0 0/0 1/0 0/1 0/0 0/0 1/0 0/1
- 28) 0/0 1/0 0/0 0/0 1/0 0/1 0/0 1/0 0/0
- 29) 0/0 1/0 0/0 0/0 1/0 0/1 1/0 0/0 0/0 0/1
- 30) 0/0 1/0 0/0 0/0 1/0 1/1 0/0 0/0 0/0
- 31) 0/0 1/0 0/0 1/0 0/0 0/0 0/0 0/0
- 32) 0/0 1/0 0/0 1/0 0/0 0/0 0/0 1/0 0/0
- 33) 0/0 1/0 0/0 1/0 0/0 0/0 1/0 0/0
- 34) 0/0 1/0 0/0 1/0 0/0 1/0 0/0 0/0 0/0
- 35) 0/0 1/0 0/0 1/0 1/0 0/0 0/0 0/0
- 36) 0/0 1/0 1/0 0/0 0/0 0/1 0/0
- 37) 0/0 1/0 1/0 0/0 0/0 0/1 1/0 0/0
- 38) 0/0 1/0 1/0 0/0 0/0 1/1 0/0
- 39) 0/0 1/0 1/0 0/0 1/0 0/1 0/0 0/0
- 40) 0/0 1/0 1/0 1/0 0/0 0/0 0/0
- 41) 1/0 0/0 0/0 0/0 0/0
- 42) 1/0 0/0 0/0 0/0 1/0 0/1
- 43) 1/0 0/0 0/0 1/0 0/0
- 44) 1/0 0/0 1/0 0/0 0/0 0/1
- 45) 1/0 1/0 0/0 0/0 0/0

ПРИЛОЖЕНИЕ В

Полный проверяющий тест $T = VW \cup VIW$, построенный для конечного автомата,
описывающего поведение логической схемы s27

- | | |
|----------------------|---------------------------|
| 1) 0/0 0/1 0/1 0/1 | 51) 1/0 9/1 |
| 2) 0/0 0/1 2/1 | 52) 2/1 0/0 0/1 |
| 3) 0/0 0/1 9/1 | 53) 2/1 2/1 |
| 4) 0/0 1/1 0/1 0/1 | 54) 2/1 9/0 |
| 5) 0/0 1/1 2/1 | 55) 3/0 0/0 0/0 0/0 |
| 6) 0/0 1/1 9/1 | 56) 3/0 0/0 2/0 |
| 7) 0/0 2/1 0/0 0/1 | 57) 3/0 0/0 9/0 |
| 8) 0/0 2/1 2/1 | 58) 3/0 1/0 0/0 0/0 |
| 9) 0/0 2/1 9/0 | 59) 3/0 1/0 2/0 |
| 10) 0/0 3/1 0/0 0/1 | 60) 3/0 1/0 9/0 |
| 11) 0/0 3/1 2/1 | 61) 3/0 2/0 0/0 0/0 |
| 12) 0/0 3/1 9/0 | 62) 3/0 2/0 2/0 |
| 13) 0/0 4/1 0/1 0/1 | 63) 3/0 2/0 9/0 |
| 14) 0/0 4/1 2/1 | 64) 3/0 3/0 0/0 0/0 |
| 15) 0/0 4/1 9/1 | 65) 3/0 3/0 2/0 |
| 16) 0/0 5/1 0/1 0/1 | 66) 3/0 3/0 9/0 |
| 17) 0/0 5/1 2/1 | 67) 3/0 4/0 0/0 0/0 0/0 |
| 18) 0/0 5/1 9/1 | 68) 3/0 4/0 0/0 2/0 |
| 19) 0/0 6/1 0/0 0/1 | 69) 3/0 4/0 0/0 9/1 |
| 20) 0/0 6/1 2/1 | 70) 3/0 4/0 1/0 0/0 0/0 |
| 21) 0/0 6/1 9/0 | 71) 3/0 4/0 1/0 2/0 |
| 22) 0/0 7/1 0/0 0/1 | 72) 3/0 4/0 1/0 9/1 |
| 23) 0/0 7/1 2/1 | 73) 3/0 4/0 2/0 0/0 0/0 |
| 24) 0/0 7/1 9/0 | 74) 3/0 4/0 2/0 2/0 |
| 25) 0/0 8/1 0/1 0/1 | 75) 3/0 4/0 2/0 9/0 |
| 26) 0/0 8/1 2/1 | 76) 3/0 4/0 3/0 0/0 0/0 |
| 27) 0/0 8/1 9/1 | 77) 3/0 4/0 3/0 2/0 |
| 28) 0/0 9/1 0/1 0/1 | 78) 3/0 4/0 3/0 9/0 |
| 29) 0/0 9/1 2/1 | 79) 3/0 4/0 4/0 0/0 0/0 |
| 30) 0/0 9/1 9/1 | 80) 3/0 4/0 4/0 2/0 |
| 31) 0/0 10/1 0/1 0/0 | 81) 3/0 4/0 4/0 9/1 |
| 32) 0/0 10/1 2/1 | 82) 3/0 4/0 5/0 0/0 0/0 |
| 33) 0/0 10/1 9/1 | 83) 3/0 4/0 5/0 2/0 |
| 34) 0/0 11/1 0/1 0/0 | 84) 3/0 4/0 5/0 9/1 |
| 35) 0/0 11/1 2/1 | 85) 3/0 4/0 6/0 0/0 0/0 |
| 36) 0/0 11/1 9/1 | 86) 3/0 4/0 6/0 2/0 |
| 37) 0/0 12/1 0/1 0/1 | 87) 3/0 4/0 6/0 9/0 |
| 38) 0/0 12/1 2/1 | 88) 3/0 4/0 7/0 0/0 0/0 |
| 39) 0/0 12/1 9/1 | 89) 3/0 4/0 7/0 2/0 |
| 40) 0/0 13/1 0/1 0/1 | 90) 3/0 4/0 7/0 9/0 |
| 41) 0/0 13/1 2/1 | 91) 3/0 4/0 8/1 0/1 0/1 |
| 42) 0/0 13/1 9/1 | 92) 3/0 4/0 8/1 2/1 |
| 43) 0/0 14/1 0/1 0/0 | 93) 3/0 4/0 8/1 9/1 |
| 44) 0/0 14/1 2/1 | 94) 3/0 4/0 9/1 0/1 0/1 |
| 45) 0/0 14/1 9/1 | 95) 3/0 4/0 9/1 2/1 |
| 46) 0/0 15/1 0/1 0/0 | 96) 3/0 4/0 9/1 9/1 |
| 47) 0/0 15/1 2/1 | 97) 3/0 4/0 10/1 0/1 0/0 |
| 48) 0/0 15/1 9/1 | 98) 3/0 4/0 10/1 2/1 |
| 49) 1/0 0/1 0/1 | 99) 3/0 4/0 10/1 9/1 |
| 50) 1/0 2/1 | 100) 3/0 4/0 11/1 0/1 0/0 |

101)	3/0 4/0 11/1 2/1	158)	7/1 2/1
102)	3/0 4/0 11/1 9/1	159)	7/1 9/0
103)	3/0 4/0 12/1 0/1 0/1	160)	8/1 0/1 0/0 0/1
104)	3/0 4/0 12/1 2/1	161)	8/1 0/1 2/1
105)	3/0 4/0 12/1 9/1	162)	8/1 0/1 9/0
106)	3/0 4/0 13/1 0/1 0/1	163)	8/1 1/1 0/0 0/1
107)	3/0 4/0 13/1 2/1	164)	8/1 1/1 2/1
108)	3/0 4/0 13/1 9/1	165)	8/1 1/1 9/0
109)	3/0 4/0 14/1 0/1 0/0	166)	8/1 2/1 0/0 0/1
110)	3/0 4/0 14/1 2/1	167)	8/1 2/1 2/1
111)	3/0 4/0 14/1 9/1	168)	8/1 2/1 9/0
112)	3/0 4/0 15/1 0/1 0/0	169)	8/1 3/1 0/0 0/1
113)	3/0 4/0 15/1 2/1	170)	8/1 3/1 2/1
114)	3/0 4/0 15/1 9/1	171)	8/1 3/1 9/0
115)	3/0 5/0 0/0 0/0	172)	8/1 4/1 0/1 0/1
116)	3/0 5/0 2/0	173)	8/1 4/1 2/1
117)	3/0 5/0 9/1	174)	8/1 4/1 9/1
118)	3/0 6/0 0/0 0/0	175)	8/1 5/1 0/1 0/1
119)	3/0 6/0 2/0	176)	8/1 5/1 2/1
120)	3/0 6/0 9/0	177)	8/1 5/1 9/1
121)	3/0 7/0 0/0 0/0	178)	8/1 6/1 0/0 0/1
122)	3/0 7/0 2/0	179)	8/1 6/1 2/1
123)	3/0 7/0 9/0	180)	8/1 6/1 9/0
124)	3/0 8/1 0/1 0/0	181)	8/1 7/1 0/0 0/1
125)	3/0 8/1 2/1	182)	8/1 7/1 2/1
126)	3/0 8/1 9/1	183)	8/1 7/1 9/0
127)	3/0 9/0 0/0 0/0	184)	8/1 8/1 0/1 0/0
128)	3/0 9/0 2/0	185)	8/1 8/1 2/1
129)	3/0 9/0 9/0	186)	8/1 8/1 9/1
130)	3/0 10/1 0/1 0/0	187)	8/1 9/1 0/1 0/0
131)	3/0 10/1 2/1	188)	8/1 9/1 2/1
132)	3/0 10/1 9/1	189)	8/1 9/1 9/1
133)	3/0 11/0 0/0 0/0	190)	8/1 10/1 0/1 0/0
134)	3/0 11/0 2/0	191)	8/1 10/1 2/1
135)	3/0 11/0 9/0	192)	8/1 10/1 9/1
136)	3/0 12/1 0/1 0/1	193)	8/1 11/1 0/1 0/0
137)	3/0 12/1 2/1	194)	8/1 11/1 2/1
138)	3/0 12/1 9/1	195)	8/1 11/1 9/1
139)	3/0 13/1 0/1 0/1	196)	8/1 12/1 0/1 0/1
140)	3/0 13/1 2/1	197)	8/1 12/1 2/1
141)	3/0 13/1 9/1	198)	8/1 12/1 9/1
142)	3/0 14/1 0/1 0/0	199)	8/1 13/1 0/1 0/1
143)	3/0 14/1 2/1	200)	8/1 13/1 2/1
144)	3/0 14/1 9/1	201)	8/1 13/1 9/1
145)	3/0 15/1 0/1 0/0	202)	8/1 14/1 0/1 0/0
146)	3/0 15/1 2/1	203)	8/1 14/1 2/1
147)	3/0 15/1 9/1	204)	8/1 14/1 9/1
148)	4/1 0/1 0/1	205)	8/1 15/1 0/1 0/0
149)	4/1 2/1	206)	8/1 15/1 2/1
150)	4/1 9/1	207)	8/1 15/1 9/1
151)	5/1 0/1 0/1	208)	9/0 0/0 0/0
152)	5/1 2/1	209)	9/0 2/0
153)	5/1 9/1	210)	9/0 9/0
154)	6/1 0/0 0/1	211)	10/1 0/1 0/0
155)	6/1 2/1	212)	10/1 2/1
156)	6/1 9/0	213)	10/1 9/1
157)	7/1 0/0 0/1	214)	11/0 0/0 0/0

215) 11/0 2/0
216) 11/0 9/0
217) 12/1 0/1 0/1
218) 12/1 2/1
219) 12/1 9/1
220) 13/1 0/1 0/1
221) 13/1 2/1

222) 13/1 9/1
223) 14/1 0/1 0/0
224) 14/1 2/1
225) 14/1 9/1
226) 15/1 0/1 0/0
227) 15/1 2/1
228) 15/1 9/1

ПРИЛОЖЕНИЕ Г

Полный проверяющий тест $T' = VI$, построенный для конечного автомата, описывающего поведение логической схемы s27

1) 0/1	39) 3/0 10/1
2) 1/0	40) 3/0 11/1
3) 2/0	41) 3/0 12/1
4) 5/1	42) 3/0 13/1
5) 6/1	43) 3/0 14/1
6) 7/1	44) 3/0 15/1
7) 8/1	45) 3/0 4/0 0/0
8) 9/0	46) 3/0 4/0 1/0
9) 10/1	47) 3/0 4/0 2/0
10) 11/0	48) 3/0 4/0 3/0
11) 12/1	49) 3/0 4/0 4/0
12) 13/1	50) 3/0 4/0 5/0
13) 14/1	51) 3/0 4/0 6/0
14) 15/1	52) 3/0 4/0 7/0
15) 4/1 0/1	53) 3/0 4/0 8/1
16) 4/1 1/1	54) 3/0 4/0 9/1
17) 4/1 2/1	55) 3/0 4/0 10/1
18) 4/1 3/1	56) 3/0 4/0 11/1
19) 4/1 4/1	57) 3/0 4/0 12/1
20) 4/1 5/1	58) 3/0 4/0 13/1
21) 4/1 6/1	59) 3/0 4/0 14/1
22) 4/1 7/1	60) 3/0 4/0 15/1
23) 4/1 8/1	61) 4/1 10/1 0/1
24) 4/1 9/1	62) 4/1 10/1 1/1
25) 4/1 11/1	63) 4/1 10/1 2/1
26) 4/1 12/1	64) 4/1 10/1 3/1
27) 4/1 13/1	65) 4/1 10/1 4/1
28) 4/1 14/1	66) 4/1 10/1 5/1
29) 4/1 15/1	67) 4/1 10/1 6/1
30) 3/0 0/0	68) 4/1 10/1 7/1
31) 3/0 1/0	69) 4/1 10/1 8/1
32) 3/0 2/0	70) 4/1 10/1 9/1
33) 3/0 3/0	71) 4/1 10/1 10/1
34) 3/0 5/0	72) 4/1 10/1 11/1
35) 3/0 6/0	73) 4/1 10/1 12/1
36) 3/0 7/0	74) 4/1 10/1 13/1
37) 3/0 8/1	75) 4/1 10/1 14/1
38) 3/0 9/1	76) 4/1 10/1 15/1



Отчет о проверке

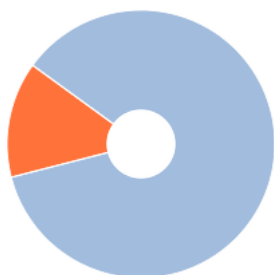
Автор: Карепин Эдуард Денисович

Название документа: Карепин

Проверяющий: Павлова Александра Андреевна

Организация: Томский Государственный Университет

РЕЗУЛЬТАТЫ ПРОВЕРКИ



Совпадения:
14,08%



Оригинальность:
85,92%



Цитирования:
0%



Самоцитирования:
0%



1

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует полному тексту проверяемого документа.



Проверено: 78,29% текста документа, исключено из проверки: 21,71% текста документа. Разделы, отключенные пользователем: Титульный лист, Содержание, Библиография, Приложение

- Совпадения** — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» — это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.
- Самоцитирования** — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» — это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.
- Цитирования** — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирования» — это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.
- Текстовое пересечение** — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
- Источник** — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
- Оригинальный текст** — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» — это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Номер документа: 29

Тип документа: Не указано

Дата проверки: 27.01.2024 15:11:21

Дата корректировки: Нет

Количество страниц: 44

Символов в тексте: 56795

Слов в тексте: 8575

Число предложений: 1312

Комментарий: не указано

ПАРАМЕТРЫ ПРОВЕРКИ

Выполнена проверка с учетом редактирования: Да

Выполнено распознавание текста (OCR): Нет

Выполнена проверка с учетом структуры: Нет

Модули поиска: Переводные заимствования*, Библиография, Издательство Wiley, Модуль поиска "tsu", Переводные заимствования (RuEn), Шаблонные фразы, Цитирование, Патенты СССР, РФ, СНГ, Кольцо вузов, СМИ России и СНГ, Переводные заимствования издательства Wiley, Переводные заимствования по Интернету (EnRu), Переводные заимствования по eLIBRARY.RU (EnRu), Сводная коллекция РГБ, Переводные заимствования по коллекции Интернет в английском сегменте, ИПС Адилет, Диссертации НББ, СПС ГАРАНТ: аналитика, Перефразированные заимствования по коллекции Интернет в английском сегменте, eLIBRARY.RU, Перефразирования по Интернету (EN), Перефразирования по eLIBRARY.RU, Коллекция НБУ, Переводные заимствования по коллекции Гарант: аналитика, СПС ГАРАНТ: нормативно-правовая документация, Перефразирования по коллекции издательства Wiley, Переводные заимствования IEEE, Перефразированные заимствования по коллекции Интернет в русском сегменте, Сводная коллекция ЭБС, Перефразирования по коллекции IEEE, Медицина, Перефразирования по СПС ГАРАНТ: аналитика, Переводные заимствования по коллекции Интернет в русском сегменте, Перефразирования по Интернету, Интернет Плюс*

ИСТОЧНИКИ

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарий
[01]	2,69%	1,5%	Вестн. Том. гос. ун-та. Управлени... http://journals.tsu.ru	15 Апр 2023	Интернет Плюс*	
[02]	2,53%	1,16%	Конечно автоматные методы ана... http://ams.tsu.ru	11 Июл 2020	Интернет Плюс*	
[03]	2,49%	0,4%	http://vital.lib.tsu.ru/vital/access/se... http://vital.lib.tsu.ru	18 Ноя 2019	Интернет Плюс*	
[04]	2,15%	0,74%	Дорофеева, Маргарита Юрьевна ... http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	
[05]	1,67%	1,67%	Разработка методов синтеза про... http://tekhnosfera.com	30 Янв 2017	Перефразирования по Интернету	
[06]	1,6%	0,96%	Прокопенко, Светлана Анатолье... http://dlib.rsl.ru	26 Дек 2011	Сводная коллекция РГБ	
[07]	1,52%	1,26%	Выходцев Диплом 2023	29 Ноя 2023	Модуль поиска "tsu"	
[08]	1,34%	0,36%	диплом.pdf	02 Июн 2022	Модуль поиска "tsu"	
[09]	1,34%	0%	ВКР Субач-2.docx	16 Янв 2022	Модуль поиска "tsu"	
[10]	1,34%	0%	Можанов В.С. Диссертация.docx	11 Июн 2023	Модуль поиска "tsu"	
[11]	1,27%	0,29%	Кирилина, преддиплом. практика	10 Мая 2021	Модуль поиска "tsu"	
[12]	1,24%	0%	thesis_Bykov	16 Июн 2023	Кольцо вузов	
[13]	1,23%	0,69%	Проверяющие эксперименты с н... http://elibrary.ru	05 Авг 2016	Перефразирования по eLIBRARY.RU	
[14]	1,22%	0%	Разработка методов синтеза про... http://tekhnosfera.com	раньше 2011	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[15]	1,14%	0%	https://vital.lib.tsu.ru/vital/access/s... https://vital.lib.tsu.ru	14 Мая 2023	Интернет Плюс*	
[16]	1,14%	1,14%	Синтез тестов для цифровых сист... http://journals.tsu.ru	28 Окт 2022	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[17]	1,08%	0%	Радиолокационные характерист...	08 Июн 2023	Модуль поиска "tsu"	
[18]	1,08%	0,43%	Посмотреть автореферат http://sun.tsu.ru	24 Дек 2023	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[19]	1,03%	0,38%	Спицына, Наталия Владимировн... http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	
[20]	1%	1%	МЕТОД СИНТЕЗА ТЕСТОВ ДЛЯ ПР... http://cyberleninka.ru	08 Янв 2017	Перефразирования по Интернету	

[21]	0,92%	0,92%	Исследование и разработка коне... http://dslib.net	01 Июл 2016	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[22]	0,87%	0%	https://www.tsu.ru/upload/medial... https://tsu.ru	12 Янв 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[23]	0,87%	0%	https://lib.tsu.ru/win/produkcija/m... https://lib.tsu.ru	19 Июн 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[24]	0,86%	0%	Синтез тестов для цифровых сист... http://elibrary.ru	11 Фев 2019	Перефразирования по eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[25]	0,78%	0%	Лапутенко, Андрей Владимирови... http://dlib.rsl.ru	01 Янв 2021	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[26]	0,72%	0%	ВКР_Будко.pdf	29 Мая 2023	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[27]	0,6%	0,6%	Логические основы проектирова... http://elibrary.ru	28 Авг 2014	Перефразирования по eLIBRARY.RU	
[28]	0,59%	0%	Методы синтеза проверяющих т... http://tekhnosfera.com	раньше 2011	Перефразированные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[29]	0,58%	0,58%	Вестн. Том. гос. ун-та. Управлени... http://journals.tsu.ru	15 Апр 2023	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[30]	0,49%	0%	Кушик, Наталья Геннадьевна дис... http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[31]	0,47%	0%	Теория и реализация языков про... http://studentlibrary.ru	19 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[32]	0,46%	0%	МЕТОДЫ СИНТЕЗА ПРОВЕРЯЮЩ... http://dissers.ru	09 Мая 2020	Перефразированные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[33]	0,43%	0%	https://srdi.tsu.ru/files/docs/srdi-2... https://srdi.tsu.ru	23 Июн 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[34]	0,42%	0%	Программная реализация автом...	27 Июн 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[35]	0,42%	0%	Программная реализация автом...	26 Июн 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[36]	0,42%	0%	ИСПОЛЬЗОВАНИЕ СВОЙСТВ МУТ... http://elibrary.ru	раньше 2011	Перефразирования по eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[37]	0,41%	0%	Методы синтеза проверяющих т... http://tekhnosfera.com	раньше 2011	Переводные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[38]	0,4%	0%	Диспетчерские центры и техноло... https://e.lanbook.com	21 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[39]	0,4%	0%	Диспетчерские центры и техноло... http://studentlibrary.ru	19 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[40]	0,38%	0%	Исследование и разработка коне... http://tekhnosfera.com	21 Окт 2014	Перефразированные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[41]	0,37%	0%	Тестирование автомата в различ... https://core.ac.uk	20 Янв 2023	Перефразированные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[42]	0,37%	0%	2_216_88_0_0.600_64764065 Посм... http://sun.tsu.ru	05 Окт 2022	Перефразированные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[43]	0,35%	0%	Метод построения полного пров...	29 Авг 2008	eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[44]	0,34%	0%	Шкурко В.В. Разработка програм...	04 Апр 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[45]	0,34%	0%	ИСПОЛЬЗОВАНИЕ СВОЙСТВ МУТ... http://elibrary.ru	раньше 2011	eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[46]	0,32%	0%	не указано	29 Сен 2022	Шаблонные фразы	Источник исключен. Причина: Маленький процент пересечения.
[47]	0,32%	0%	Презентация на тему: "Теория вы... http://myshared.ru	27 Янв 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[48]	0,31%	0%	Andriyanova_edit.docx	16 Июн 2020	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.

[49]	0,31%	0%	ВКР.docx	07 Июн 2023	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[50]	0,31%	0%	Выпускная квалификационная ра...	11 Июн 2023	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[51]	0,31%	0%	Васильева Анастасия Константин...	09 Июн 2023	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[52]	0,31%	0%	Термодинамические условия фо...	08 Июн 2023	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[53]	0,31%	0%	Оптимизация налогообложения ...	24 Ноя 2020	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[54]	0,31%	0%	Онскуль_02.06.docx	02 Июн 2021	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[55]	0,31%	0%	Diplom_Kovalenko_031915 (1)	09 Июн 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[56]	0,3%	0%	Разработка компиляторов https://book.ru	03 Июл 2017	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[57]	0,29%	0%	https://core.ac.uk/download/pdf/3... https://core.ac.uk	21 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[58]	0,28%	0%	Модели безопасности компьюте... http://studentlibrary.ru	20 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[59]	0,28%	0%	https://www.tsu.ru/upload/mediali... https://tsu.ru	24 Янв 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[60]	0,27%	0%	не указано https://edu.tusur.ru	27 Янв 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[61]	0,26%	0%	http://abit.mstu.edu.ru/docs/files/a... http://abit.mstu.edu.ru	22 Июн 2020	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[62]	0,26%	0%	http://abit.mstu.edu.ru/docs/files/a... http://abit.mstu.edu.ru	16 Фев 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[63]	0,26%	0%	http://abit.mstu.edu.ru/docs/files/a... http://abit.mstu.edu.ru	20 Июн 2019	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[64]	0,24%	0%	Буфалов, Сергей Анатольевич ди... http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[65]	0,24%	0%	Синтез тестов для цифровых сист... http://elibrary.ru	11 Фев 2019	eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[66]	0,23%	0%	Проверяющие эксперименты с н... http://elibrary.ru	05 Авг 2016	eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[67]	0,22%	0%	http://ui.tsu.ru/wp-content/upload... http://ui.tsu.ru	24 Авг 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[68]	0,17%	0%	ISBN5922106309.txt	26 Окт 2017	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.