

УДК 004.312.4

СТРУКТУРНЫЕ МОДЕЛИ ДЛЯ ОБНАРУЖЕНИЯ НЕИСПРАВНОСТЕЙ КОНЕЧНЫХ АВТОМАТОВ МУРА¹© 2023 г. В. В. Соловьев^{а,*}^аБелостокский технологический ун-т, Белосток, Польша

*e-mail: valsol@mail.ru

Поступила в редакцию 30.01.2023 г.

После доработки 08.06.2023 г.

Принята к публикации 31.07.2023 г.

Обнаружение неисправностей является важной задачей при проектировании отказоустойчивых конечных автоматов. Предлагаются структурные модели конечных автоматов Мура для обнаружения многократных неисправностей в различных элементах конечного автомата и предотвращения их негативного воздействия на управляемый объект. Рассматриваемые структурные модели позволяют обнаруживать недопустимые входные и выходные векторы как в каждом состоянии, так и для всего автомата, недопустимый код настоящего и следующего состояния автомата, а также недопустимые переходы между состояниями. Издержки реализации предлагаемых структур по площади в среднем составляют от 3 до 26%, а быстродействие автомата либо не изменяется, либо даже увеличивается в среднем на 24–30%. Приводятся оценки площади и быстродействия предлагаемых структурных моделей конечных автоматов, даются рекомендации для их практического использования. Показано, что выбор подходящей структуры позволяет не увеличивать площадь, а в некоторых случаях даже приводит к возрастанию быстродействия конечного автомата.

DOI: 10.31857/S0002338823060100, EDN: GSVСХМ

Введение. В настоящее время беспилотные летательные аппараты (БПЛА) широко используются во многих сферах деятельности человека, в том числе и в военных конфликтах. Одним из способов борьбы с БПЛА является воздействие на БПЛА электромагнитного импульса с целью вывода из строя устройства управления [1]. Эффективным способом защиты БПЛА от воздействия электромагнитного импульса служит реализация устройства управления в виде отказоустойчивого конечного автомата.

На рис. 1 показана традиционная структурная модель конечного автомата, где X – входные сигналы (входной вектор); Y – выходные сигналы (выходной вектор); R – регистр состояний, в котором хранится код настоящего (текущего) состояния; Φ – комбинационная схема (логика), определяющая код следующего состояния ($\Phi: X \times S \rightarrow S$); Ψ – комбинационная схема, формирующая значения выходных сигналов ($\Psi: X \times S \rightarrow Y$); S – множество состояний конечного автомата. Отметим, что выход регистра R соединен цепью обратной связи с входом комбинационной схемы Φ . Для автоматов Мура значения выходных сигналов формируются на основании кода настоящего состояния ($\Psi: S \rightarrow Y$), а для автоматов Мили – на базе кода настоящего состояния и значений входных сигналов ($\Psi: X \times S \rightarrow Y$). Поэтому для автомата Мили входы X соединены с входами комбинационной схемы Ψ (на рис. 1 обозначено пунктирной линией). Тактовый сигнал clk для регистра R формируется с помощью генератора *Oscillator*.

В результате воздействия электромагнитного импульса сбои в структуре конечного автомата на рис. 1 могут возникать:

- на входах: недопустимый входной вектор X ;
- в регистре R : код недопустимого настоящего состояния;
- в логике Φ : формирование недопустимого кода следующего состояния;
- в логике Ψ : формирование недопустимого выходного вектора;

¹ Работа выполнена при финансовой поддержке Белостокского технологического университета (Польша) (грант № WZ/WI-III/5/2023).

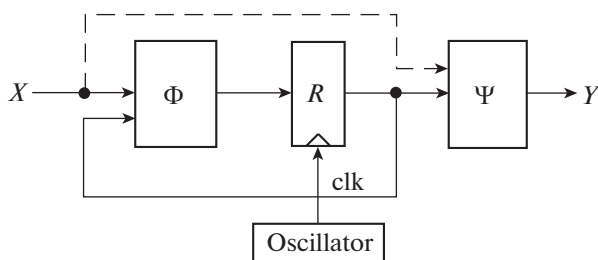


Рис. 1. Традиционная структурная модель конечных автоматов

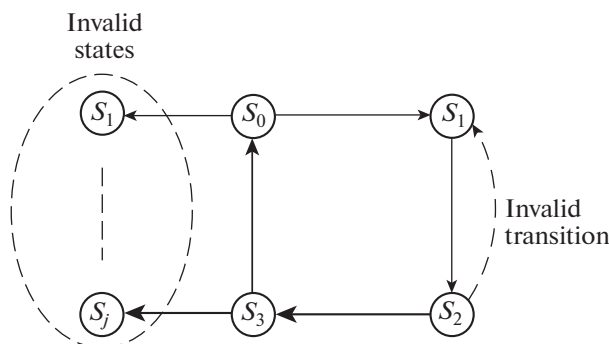


Рис. 2. Возможные неисправности конечного автомата

в цепях конечного автомата: входных, выходных, обратной связи;

в цепи синхросигнала clk : отсутствие тактового сигнала;

в генераторе Oscillator: выход из строя генератора тактовых сигналов.

Кроме того, в результате воздействия электромагнитного импульса возможны неисправности автомата на поведенческом уровне: переход автомата в недопустимые состояния (invalid states), а также недопустимые переходы (invalid transition) в допустимые состояния (рис. 2).

Известно достаточно много методов построения отказоустойчивых конечных автоматов. Большинство из них посвящены противодействию одиночным неисправностям (single event upsets – SEUs), порождаемых радиационным излучением и космическими лучами, которые изменяют содержимое триггеров или ячеек памяти, при этом первичные входы всегда считаются верными. Однако электромагнитный импульс характеризуется следующими особенностями воздействия на БПЛА:

значительная продолжительность времени воздействия по сравнению с космической частицей;

воздействие сразу на все компоненты схемы;

порождает не кратковременные одиночные неисправности (SEU), а продолжительные многократные ошибки;

воздействует главным образом на цепи (входные, выходные, обратной связи);

редко изменяет содержимое регистра или ячеек памяти.

Отметим, что современные методики проектирования цифровых устройств [2] значительно отличаются от подходов, используемых несколько десятилетий назад. В последнее время значительное развитие получили языки описания аппаратуры (hardware description language – HDL), такие, как, VHDL, Verilog, SystemVerilog и др. В результате синтез устройства сводится к корректному описанию проекта на некотором языке HDL [3]. При этом отпадают традиционные этапы синтеза конечного автомата: кодирование состояний; генерация логических (булевых) уравнений для комбинационных схем Φ и Ψ ; минимизация, факторизация и декомпозиция логики. Все эти действия средства синтеза выполняют автоматически. Однако в результате выполнения автоматической оптимизации логики избыточная логика, направленная на построение отказо-

устойчивого конечного автомата, может быть удалена из проекта. Поэтому необходимы новые подходы к проектированию отказоустойчивых конечных автоматов.

Областью исследований данной работы является проектирование отказоустойчивых конечных автоматов. Но прежде чем исправить каким-либо образом сбой конечного автомата, неисправность необходимо обнаружить. Поскольку в практике инженерного проектирования наиболее часто используется модель автомата Мура, в настоящей работе исследуются структурные модели для обнаружения неисправностей конечных автоматов Мура.

В статье впервые решается задача обнаружения неисправностей конечного автомата на структурном уровне. Для этого предложены структурные модели конечных автоматов, которые позволяют обнаружить многократные неисправности в различных элементах автомата и предотвратить их негативное воздействие на управляемый объект. Новизна структурных моделей заключается в следующем:

впервые рассматриваются неисправности конечного автомата, которые вызваны не космическими частицами, а электромагнитным импульсом, воздействующим на устройство управления БПЛА;

не ограничено число неисправностей в кодах состояний, во входном и выходном векторах;

структурные модели позволяют обнаруживать неисправности не только в регистре состояний конечного автомата, но также во входном векторе, в логике формирования кода следующего состояния, в логике формирования выходных сигналов, а также в цепи обратной связи;

кроме того, структурные модели выявляют недопустимые переходы автомата и переходы в недопустимые состояния;

структурные модели не только находят неисправности, но и предотвращают их негативное воздействие на управляемый объект;

объединенная структурная модель конечного автомата позволяет одновременно (параллельно) определять неисправности сразу во всех элементах конечного автомата;

структурные модели применимы при реализации конечного автомата как в программируемых интегральных схемах (ПЛИС – field programmable gate arrays – FPGAs), так и в специализированных интегральных схемах (application-specific integrated circuit – ASIC).

Главной целью работы является решение на структурном уровне задачи обнаружения многократных неисправностей различных компонентов конечного автомата Мура, которые могут быть вызваны электромагнитным импульсом.

Статья организована следующим образом. В разд. 1 рассматриваются публикации, относящиеся к проблеме проектирования отказоустойчивых конечных автоматов. В разд. 2 описывается пример автомата Мура, который используется для объяснения предлагаемого подхода. В разд. 3 представлены структурные модели конечных автоматов. Экспериментальные исследования исследуются в разд. 4. В разд. 5 приведены оценки площади и быстродействия рассматриваемых структурных моделей. В разд. 6 на конкретных примерах даны рекомендации по практическому применению структурных моделей. В Заключение подводятся итоги и указывается на направления дальнейших исследований.

1. Относящиеся работы. Проектирование отказоустойчивых вычислительных и управляющих систем, которыми являются конечные автоматы, всегда было актуальным с момента появления компьютеров. Проблема защиты конечных автоматов от космических лучей интенсивно исследовалась в конце 60-х годов прошлого столетия. Традиционным решением задачи считается кратное дублирование архитектуры конечного автомата, при этом наибольшее распространение получил метод тройного модульного резервирования (triple modular redundancy – TRM) для защиты от SEU [4]. В общем случае отказоустойчивость цифровой системы может обеспечиваться избыточностью архитектуры, увеличением времени работы и избыточностью данных [5].

Большое число работ посвящено улучшению метода TMR. Для этого в [6] используется код с исправлением одиночных ошибок (single error correction – SEC) для реализации конечного автомата, который допускает одну ошибку в логике следующего состояния и в регистре состояний. В [7] сравниваются архитектуры отказоустойчивых конечных автоматов (TRM, SEU-ITRM, duplex, EEC, модифицированная EEC и IEC), которые допускают единичные ошибки при переключении состояний.

Применение ПЛИС для проектирования конечных автоматов дает ряд преимуществ по сравнению с ASIC: низкая потребляемая мощность, небольшая стоимость, малое время выхода готового изделия на рынок, возможность перепрограммирования и др. Однако ПЛИС более подвер-

жены SEU, вызываемые космическими частицами, чем ASIC. Поэтому фирма Xilinx выпустила специальную серию ПЛИС семейства Virtex, которая поддерживает метод TMR на аппаратном уровне [8]. В [9] предлагается в системах на кристалле (system-on-chip – SoC) при обнаружении ошибки в ПЛИС генерировать прерывание для микроконтроллера, который запускает процедуру частичного перепрограммирования ПЛИС. В [10] приводится метод усовершенствования TMR, который сочетает в себе дублирование со сравнением (duplication with comparison – DWC) и одновременное обнаружение ошибок (concurrent error detection – CED) на основе временной избыточности.

Отдельные работы посвящены способам кодирования состояний отказоустойчивых конечных автоматов. В [11] рассматриваются четыре способа кодирования состояний отказоустойчивых конечных автоматов: binary, one-hot, Хэмминга с расстоянием 2 (H2) и Хэмминга с расстоянием 3 (H3). Сравняется обеспечение отказоустойчивости и использование ресурсов. В [12] исследуются способы кодирования состояний binary, one-hot и H3 для устранения SEU в регистре состояний, рекомендуется вручную задавать логику восстановления из недопустимого состояния.

Было разработано много методов улучшения TMR, при этом используются различные способы кодирования состояний. В [13] предлагается метод двойной модульной избыточности (dual modular redundancy – DMR) и применение бита четности при реализации конечного автомата во встроенных блоках памяти FPGA. В [14] оцениваются два метода синтеза отказоустойчивых конечных автоматов: дублирование с самопроверкой и TMR. При этом используются следующие способы кодирования состояний: binary, one-hot и Грэя. В [15] предлагается улучшение метода TMR с помощью кода Хэмминга при реализации конечного автомата во встроенных блоках памяти ПЛИС. В [16] предлагается усовершенствование метода TMR от Xilinx [8]. Для этого система представляется как совокупность конечных автоматов, вводятся контрольные точки для определения неисправного домена. Когда обнаруживается неисправность, восстанавливается только неисправный домен, а остальная часть системы продолжает работать. В результате сокращается время восстановления системы после сбоя.

В некоторых методах в конечный автомат вводятся дополнительные состояния. В [17] рассматривается метод защиты конечного автомата от SEU. С этой целью в конечный автомат добавляются избыточные эквивалентные состояния для защиты состояний с высокой вероятностью возникновения сбоя. В [18] приводится метод синтеза отказоустойчивых конечных автоматов на основе кода исправления одиночных ошибок и обнаружения двойных ошибок (single error correction and double error detection – SEC-DED). Метод предусматривает возврат конечного автомата в известное безопасное состояние или состояние сброса.

В последнее время интерес к проектированию отказоустойчивых конечных автоматов по-прежнему не ослабевает. В [19] исследуется отказоустойчивость трех методов синтеза конечных автоматов: TMR, H3 и безопасный (safe) синтез. В [20] предлагаемый метод повышает отказоустойчивость конечного автомата за счет выборочного применения метода TMR в соответствии с важностью состояния. В [21] описывается квазичувствительная к задержке архитектура конечного автомата, которая сравнивается с TMR.

Рассмотрение вышеупомянутых работ показывает, что большинство методов синтеза отказоустойчивых конечных автоматов посвящено усовершенствованию метода TMR для исправления SEU в регистре состояний. При этом предполагается, что первичные входы, логика формирования кода следующего состояния, логика формирования выходов, а также дополнительная логика обнаружения и исправления ошибок сбоев не имеют.

В то же время отличные от традиционных структурные модели конечных автоматов являются очень эффективными для повышения быстродействия, уменьшения стоимости реализации и потребляемой мощности при реализации конечных автоматов на ПЛИС [22]. В настоящей работе исследуются структурные модели конечных автоматов Мура для обнаружения многократных неисправностей в различных элементах конечного автомата и предотвращения их негативного воздействия на управляемый объект.

2. Демонстрационный пример. В качестве примера рассмотрим конечный автомат Мура, граф которого показан на рис. 3. Наш автомат имеет пять состояний, три входа и три выхода. Вершины графа соответствуют состояниям S_0, \dots, S_4 , а дуги графа – переходам автомата. Вблизи каждой дуги графа записывается входной вектор, который инициирует данный переход. Вблизи каждой вершины графа записывается выходной вектор, который формируется в данном состоянии. Здесь дефис (“–”) может принимать любое значение бита: 0 или 1.

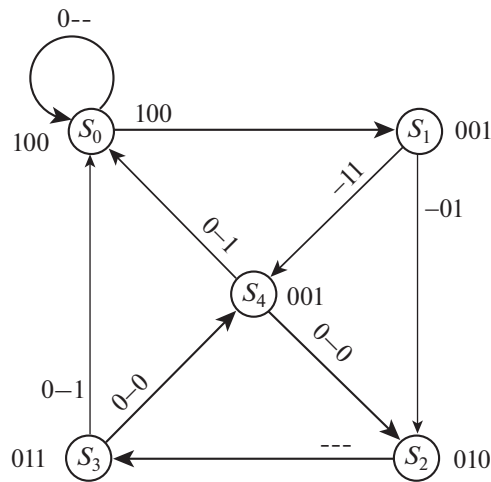


Рис. 3. Граф конечного автомата Мура

Допустимые переходы между состояниями (табл. 1) и допустимые выходные векторы в каждом состоянии (табл. 2) можно определить непосредственно по графу автомата. Допустимые входные векторы в каждом состоянии указывает разработчик на основании поведения устройства. Пусть для нашего автомата допустимые входные векторы заданы с помощью табл. 3.

3. Структурные модели конечных автоматов для обнаружения неисправностей. Структурные модели конечных автоматов для обнаружения неисправностей и предотвращения негативного воздействия неисправностей на управляемый объект показаны на рис. 4, где TVI – структура для обнаружения недопустимых входных векторов для всего автомата; VI – структура для обнаружения недопустимых входных векторов в каждом состоянии; VS – структура для обнаружения недопустимого кода настоящего состояния; VNS – структура для обнаружения недопустимого кода следующего состояния; VT – структура для обнаружения недопустимых переходов; TVO – структура для обнаружения недопустимых выходных векторов для всего автомата; VO – структура для обнаружения недопустимых выходных векторов в каждом состоянии.

Структуры на рис. 4 образованы путем введения в традиционную структуру на рис. 1 следующих комбинационных схем: TVI (total valid inputs), VI (valid inputs), VS (valid states), VNS (valid next states), VT (valid transitions), TVO (total void outputs) и VO (void outputs). Каждая дополнительная комбинационная схема формирует на выходе одноименный диагностический сигнал, единичное значение которого указывает на отсутствие неисправности. Все диагностические сигналы tv_i, v_i, vs, vns, vt, tvo и vo управляют входом разрешения синхронизации CE (clock enable) регистра состояний R. Поэтому в случае обнаружения неисправности конечный автомат будет оставаться в состоянии state до устранения неисправности. Кроме того, в структурах VS, TVO и

Таблица 1. Допустимые переходы конечного автомата

Состояние	
настоящее	следующее
S ₀	S ₀
	S ₁
S ₁	S ₂
	S ₄
S ₂	S ₃
S ₃	S ₀
	S ₄
S ₄	S ₀
	S ₂

Таблица 2. Допустимые выходные векторы конечного автомата в каждом состоянии

Состояние	Выходной вектор
S_0	100
S_1	001
S_2	010
S_3	011
S_4	001

Таблица 3. Допустимые входные векторы конечного автомата в каждом состоянии

Состояние	Входные векторы
S_0	---
S_1	001 101 011 111
S_2	---
S_3	000 010 001 011
S_4	000 010 001 011

VO дополнительно установлен выходной регистр R_O , вход CE которого управляется сигналами vs , tvo и vo соответственно. В случае обнаружения неисправности регистр R_O позволяет сохранить на выходах автомата ранее сформированный выходной вектор Y . Отметим, что введение в структуру конечного автомата выходного регистра R_O приводит к задержке выходных сигналов автомата на один такт синхронизации. Структуры TVI, VI, VNS и VT не имеют выходного вектора R_O , поскольку обнаруживаемые этими структурами неисправности не изменяют значение выходного вектора Y . Поэтому для устранения негативного воздействия неисправностей, обнаруживаемых структурами TVI, VI, VNS и VT, достаточно остановить конечный автомат в настоящем состоянии state.

На входы каждой дополнительной комбинационной схемы поступают значения, необходимые для определения соответствующей неисправности. Так, на вход комбинационной схемы TVI поступает входной вектор X . Недопустимые входные векторы для всего автомата определяются разработчиком на основании поведения управляемого объекта. Пусть для нашего примера недопустимыми входными векторами являются 110 и 111. На вход комбинационной схемы VI поступает код настоящего состояния state, а также входной вектор X . Для нашего примера функционирование комбинационной схемы VI определено в табл. 3. На вход комбинационной схемы VS поступает код настоящего состояния state. На вход комбинационной схемы VNS поступает сформированный логикой Φ код следующего состояния next. Для нашего примера допустимыми являются коды состояний S_0, \dots, S_4 . На вход комбинационной схемы VT поступает код настоящего состояния state, а также код следующего состояния next. Для нашего примера функционирование комбинационной схемы VT представлено в табл. 1. На вход комбинационной схемы TVO поступают значения выходов логики Ψ . Для нашего примера допустимыми являются выходные векторы, которые формируются в состояниях автомата: 100, 001, 010, 011 и 001. На вход комбинационной схемы VO поступает код настоящего состояния state и значения выходов, сформированные логикой Ψ . Для нашего примера функционирование комбинационной схемы VO приведено в табл. 2.

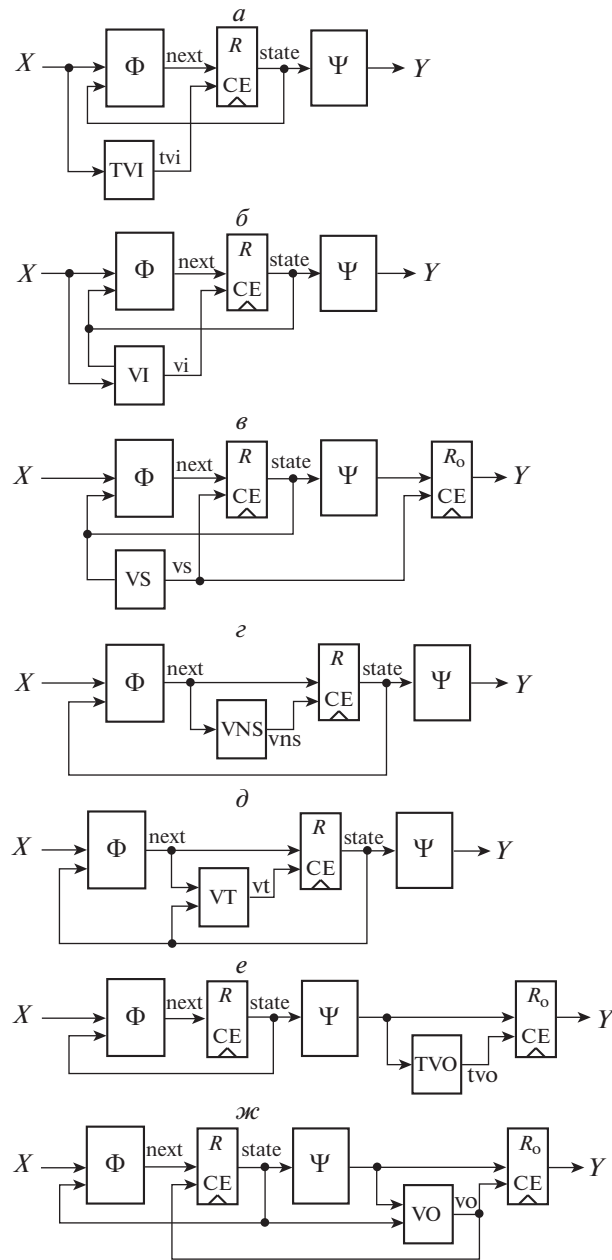


Рис. 4. Структурные модели для обнаружения неисправностей: а – структура TVI; б – структура VI; в – структура VS; г – структура VNS; д – структура VT; е – структура TVO; ж – структура VO

В табл. 4 приведены возможные причины неисправностей, которые обнаруживаются структурными моделями на рис. 4, где Φ – сбой в логике Φ ; X – недопустимый входной вектор X ; *feedback* – сбой в цепи обратной связи; Ψ – сбой в логике Ψ ; R – сбой в регистре состояний R .

Все рассмотренные структурные модели конечных автоматов не нацелены для реализации на определенную элементную базу: каждая структурная модель может быть реализована как на ASIC, так и на ПЛИС.

Структуры на рис. 4 по желанию пользователя могут произвольно объединяться вместе для наиболее эффективного обнаружения неисправностей автомата. Отметим, что структура TVI покрывается структурой VI, т.е. в случае обнаружения неисправности с помощью структуры TVI, эта неисправность обязательно будет обнаружена структурой VI. Аналогично структура TVO покрывается структурой VO, а структура VNS – структурой VT. Поэтому при объединении всех рас-

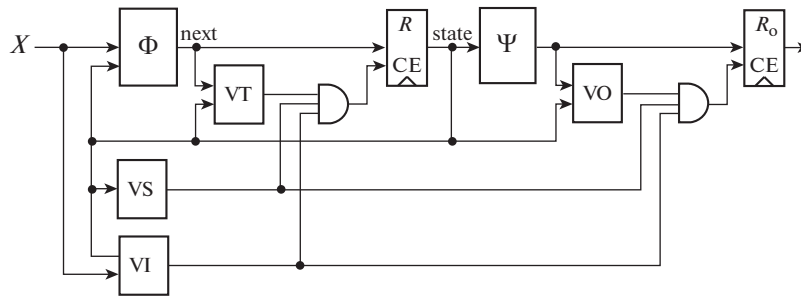


Рис. 5. Обобщенная структурная модель VISTO для обнаружения неисправностей конечных автоматов Мура

смаатриваемых структур вместе структуры TVI, TVO и NVS можно опустить. На рис. 5 показана обобщенная структура VISTO, которая объединяет структуры VI, VS, VT и VO.

Диагностические сигналы tv_i , vi , vs , vns , vt , tvo и vo , формируемые в структурных моделях конечных автоматов на рис. 4, могут выводиться на внешние выходы автомата для обнаружения местоположения неисправности в ASIC. При необходимости диагностические сигналы могут объединяться вместе для указания на ошибку (рис. 6), например, с целью реконфигурирования ПЛИС.

4. Экспериментальные исследования. Все рассмотренные структурные модели для нашего примера конечного автомата Мура были описаны на языке Verilog в стиле с тремя процессами [3]. В результате были созданы следующие проекты конечного автомата Мура для нашего примера: *based_FSM* – базовая структурная модель конечного автомата Мура (рис. 1); *TVI_FSM* – конечный автомат с проверкой допустимых входных векторов для всего автомата (рис. 4,а); *VI_FSM* – конечный автомат с проверкой допустимых входных векторов в каждом состоянии (рис. 4,б); *VS_FSM* – конечный автомат с проверкой допустимого кода настоящего состояния (рис. 4,в); *VNS_FSM* – конечный автомат с проверкой допустимого кода следующего состояния (рис. 4,г); *VT_FSM* – конечный автомат с проверкой допустимых переходов между состояниями (рис. 4,д); *TVO_FSM* – конечный автомат с проверкой допустимых выходных векторов для всего автомата (рис. 4,е); *VO_FSM* – конечный автомат с проверкой допустимых выходных векторов в каждом состоянии (рис. 4,ж); *VISTO_FSM* – конечный автомат с объединением всех рассмотренных проверок (рис. 5).

Для оценки параметров предлагаемых структурных моделей все проекты были реализованы с помощью средства Quartus Prime версии 22.4 на ПЛИС семейства Cyclone 10 LP. Результаты экспериментальных исследований приведены в табл. 5, где L – число используемых LUT (look-up table) или площадь; F – максимальная частота функционирования (в мегагерцах) или быстродействие; L_b и F_b – аналогичные параметры для базовой структурной модели автомата Мура; L/L_b и F/F_b – отношения соответствующих параметров.

Анализ табл. 5 показывает, что по стоимости реализации (площади) наиболее затратной является структура VISTO, которая приводит к увеличению площади на 109%. Затем по убыванию используемой площади следуют структуры VT (увеличение площади на 73%), VI (увеличение площади на 36%) и TVI (увеличение площади на 9%). Отметим, что структуры VO, VS, VNS и TVO не приводят к увеличению площади ПЛИС по сравнению с базовой структурой.

Таблица 4. Возможные причины неисправностей, обнаруживаемые структурными моделями конечных автоматов

Причины неисправности	TVI	VI	VS	VNS	VT	TVO	VO
Φ				*	*		
X	*	*		*	*		
Feedback		*	*	*	*		*
Ψ						*	*
R		*	*	*	*	*	*

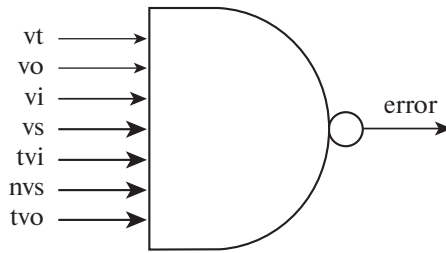


Рис. 6. Схема обнаружения ошибки конечного автомата

По быстродействию наиболее затратной также является структура VISTO, которая приводит к уменьшению быстродействия на 38%. Затем следуют структуры VT (уменьшение быстродействия на 35%) и TVI (уменьшение быстродействия на 8%). Кроме того, для структур VO, VS, TVO и VISTO выходные сигналы формируются с задержкой на один такт синхронизации по сравнению с базовой структурой. В то же время для структур VS и TVO быстродействие увеличивается на 36%, а для структуры VO – на 28%. Отметим также, что для структуры VI, несмотря на увеличение площади на 36%, быстродействие увеличивается на 10%, при этом отсутствует задержка выходных сигналов на один такт синхронизации. Структура VNS по площади и по быстродействию совпадает с базовой структурой.

Рассмотрение одного примера конечного автомата не позволяет в полной мере оценить предлагаемые структурные модели. Поэтому были проведены исследования структурных моделей на эталонных примерах конечных автоматов центра MCNC [23]. Результаты таких исследований представлены в табл. 6 и 7.

В табл. 6 приведены результаты сравнения площади эталонных конечных автоматов в случае использования предлагаемых структурных моделей по сравнению с базовой структурой, где i – число входов; o – число выходов; s – число состояний конечного автомата; C_b – площадь (число LUT) базовой структуры конечного автомата; C – площадь реализации конечного автомата в случае применения одной из предложенных структурных моделей; C_b/C – отношение соответствующих параметров; mid – среднееарифметическое значение параметра. Эталонные примеры конечных автоматов Мили предварительно были приведены к конечным автоматам Мура с помощью метода [24].

Отметим, что для структур VS и VNS площадь реализации совпадает.

Таблица 6 показывает, что наименее затратными по площади являются структуры TVO и TVI (увеличение площади в среднем на 3%). Однако структура TVI позволяет определить недопустимые входные векторы для всего конечного автомата, которых может вообще не быть, поскольку большинство конечных автоматов не имеют ограничений на входные векторы. Недопустимых выходных векторов для всего конечного автомата, которые определяет структура TVO, тоже может быть немного или вообще не быть. Следующими по аппаратным затратам следуют структуры VS и VNS (увеличение площади в среднем на 12%), затем следуют структуры VO и VT (увеличение

Таблица 5. Результаты экспериментальных исследований структурных моделей для нашего примера конечного автомата Мура

Проект	L	L/L_b	F	F/F_b	Примечание
Based_FSM	11	1	355	1	
TVI_FSM	12	1.09	328	0.92	
VI_FSM	15	1.36	389	1.10	
VS_FSM	11	1	482	1.36	Задержка на один такт
VNS_FSM	11	1	355	1	
VT_FSM	19	1.73	229	0.65	
TVO_FSM	11	1	482	1.36	Задержка на один такт
VO_FSM	11	1	453	1.28	То же
VISTO_FSM	23	2.09	220	0.62	»

Таблица 6. Сравнение площади (в числе LUT) базовой структуры конечного автомата с предложенными структурными моделями

FSM	i	o	s	C_b	VS, VNS		TVO		VO		VT		VI		TVI	
					C	C/C_b	C	C/C_b	C	C/C_b	C	C/C_b	C	C/C_b	C	C/C_b
Bbase	7	7	24	44	53	1.20	46	1.05	54	1.23	60	1.36	54	1.23	46	1.05
Cse	7	7	29	103	112	1.09	105	1.02	115	1.12	122	1.18	115	1.12	105	1.02
Ex1	9	19	78	256	280	1.09	262	1.02	288	1.13	308	1.20	285	1.11	259	1.01
Ex2	2	2	23	47	55	1.17	48	1.02	55	1.17	62	1.32	55	1.17	48	1.02
Ex3	2	2	13	31	34	1.10	32	1.03	36	1.16	40	1.29	36	1.16	32	1.03
Ex5	2	2	16	30	35	1.17	31	1.03	36	1.20	41	1.37	36	1.20	31	1.03
Ex6	5	8	14	59	63	1.08	62	1.05	65	1.10	68	1.15	65	1.10	61	1.03
Ex7	2	2	14	30	34	1.13	31	1.03	35	1.17	39	1.30	35	1.17	31	1.03
Keyb	7	2	20	66	72	1.09	67	1.02	73	1.11	79	1.20	75	1.14	68	1.03
Planet	7	19	95	173	210	1.21	179	1.03	211	1.22	236	1.36	207	1.20	175	1.01
Pma	8	8	49	153	168	1.10	156	1.02	172	1.12	186	1.22	172	1.12	156	1.02
S386	7	7	23	50	57	1.14	52	1.04	60	1.20	65	1.30	60	1.20	52	1.04
Sand	11	9	84	213	237	1.11	215	1.01	244	1.15	269	1.26	245	1.15	217	1.02
Styr	9	10	57	165	181	1.10	168	1.02	187	1.13	203	1.23	187	1.13	168	1.02
Tma	7	6	38	118	129	1.09	120	1.02	133	1.13	143	1.21	133	1.13	120	1.02
Mid						1.12		1.03		1.16		1.26		1.16		1.03

Таблица 7. Сравнение быстродействия базовой структуры конечного автомата с предложенными структурными моделями

Эталонный конечный автомат	F_b	TVI, VI, VNS, VT		VS		TVO		VO	
		F	F/F_b	F	F/F_b	F	F/F_b	F	F/F_b
Bbase	530	530	1	648	1.22	651	1.23	635	1.20
Cse	237	237	1	311	1.31	315	1.33	298	1.26
Ex1	211	211	1	309	1.46	307	1.45	305	1.45
Ex2	433	433	1	567	1.31	571	1.32	554	1.28
Ex3	445	445	1	593	1.33	592	1.33	543	1.22
Ex5	493	493	1	611	1.24	611	1.24	588	1.19
Ex6	381	381	1	495	1.30	498	1.31	472	1.24
Ex7	376	376	1	467	1.24	471	1.25	463	1.23
Keyb	459	459	1	585	1.27	581	1.27	562	1.22
Planet	589	589	1	684	1.16	682	1.16	663	1.13
Pma	346	346	1	431	1.26	442	1.28	410	1.18
S386	456	456	1	594	1.30	592	1.30	575	1.26
Sand	280	280	1	376	1.34	374	1.34	365	1.30
Styr	248	248	1	327	1.32	332	1.34	314	1.27
Tma	118	118	1	162	1.37	164	1.39	137	1.16
Mid					1.30		1.30		1.24

площади в среднем на 16%), а самой затратной по площади является структура VT (увеличение площади в среднем на 26%).

В табл. 7 приведены результаты сравнения быстродействия конечных автоматов в случае использования предлагаемых структурных моделей по сравнению с базовой структурой, где F_b – максимальная частота функционирования (в мегагерцах) базовой структуры конечного автомата;

Таблица 8. Ранги дополнительных комбинационных схем для диагностики неисправностей конечного автомата

Комбинационная схема	Ранг
TVI	$r = i$
VI	$r = i + k$
VS	$r = k$
VNS	$r = k$
VT	$r = 2k$
TVO	$r = o$
VO	$r = o + k$
VISTO	$r = i + o + 4k$

F – максимальная частота функционирования конечного автомата в случае применения одной из предложенных структурных моделей; F/F_q – отношение соответствующих параметров; mid – среднеарифметическое значение параметра.

Таблица 7 показывает, что быстродействие структур TVI, VI, VNS, VT полностью совпадает с быстродействием базовой структуры конечного автомата. Быстродействие структур VS и TVO возрастает в среднем на 30% по сравнению с базовой структурой, а быстродействие структуры VO возрастает в среднем на 24% по сравнению с базовой структурой конечного автомата.

5. Оценка площади и быстродействия предлагаемых структурных моделей. В случае реализации конечного автомата в ПЛИС на основе LUT площадь дополнительной комбинационной схемы можно определить с помощью следующего выражения:

$$C = \begin{cases} 1 & \text{при } r \leq n, \\ \left\lfloor \frac{r-n}{n-1} \right\rfloor + 1 & \text{при } r > n, \end{cases} \quad (5.1)$$

где n – число входов LUT; r – ранг (число входов) дополнительных комбинационных схем; $\lfloor A \rfloor$ – наибольшее целое число, большее или равное числу A .

Для рассматриваемых структурных моделей ранги дополнительных комбинационных схем приведены в табл. 8, где i – число входов; o – число выходов; k – число битов кодов состояний конечного автомата.

Отметим, что ранги дополнительных комбинационных схем зависят от числа k битов кодов состояний конечного автомата, т.е. от способа кодирования состояний. Поэтому, например, площадь реализации предлагаемых структурных моделей будет разной в случае использования кодирования one-hot и binary.

Оценка стоимости реализации предлагаемых структурных моделей для нашего примера представлена в табл. 9, где r – ранг дополнительной схемы (или схем) в случае кодирования one-hot; C_c – вычисленная площадь (в числе LUT) дополнительной комбинационной схемы (или схем); C_q – реальная площадь реализации, определенная с помощью средства Quartus; C_b – площадь базовой структуры конечного автомата. Отметим, что для нашего примера конечного автомата Мура в случае кодирования one-hot имеем $i = 3$, $o = 3$, $k = 5$ и $C_b = 11$.

Таблица 9 показывает, что на реальную площадь реализации C_q значительное влияние оказывает используемое средство синтеза. Поэтому вычисленное значение площади ($C_c + C_b$) может быть как больше, так и меньше реальной площади реализации. Например, для нашего примера вычисленное значение площади для большинства структур превышает или равно реальной площади реализации, исключениями являются структуры VT и VISTO.

Площадь дополнительных комбинационных схем, вычисленная, согласно (5.1), может применяться для выбора наиболее подходящей структурной модели среди других предлагаемых структурных моделей обнаружения неисправностей конечного автомата.

Таблица 9. Сравнение расчетной ($C_c + C_b$) и реальной (C_q) площади реализации нашего конечного автомата в случае кодирования one-hot

Структура	r	C_c	$C_c + C_b$	C_q
TVI	3	1	12	12
VI	8	3	14	14
VS	5	2	13	11
VNS	5	2	13	11
VT	10	3	14	19
TVO	5	2	13	11
VO	8	3	14	11
VISTO	26	10	21	23

Таблица 10. Свойства структурных моделей конечных автоматов

Свойство	TVI	VI	VS	VNS	VT	TVO	VO
Не задерживает на такт выходные сигналы	*	*		*	*		
Не увеличивает площадь	*		*	*		*	
Увеличивает быстродействие			*			*	*

Быстродействие дополнительных комбинационных схем в случае реализации конечного автомата на ПЛИС, основанных на LUT, можно оценить с помощью числа l уровней LUT, необходимых для их реализации [25]:

$$l = \lceil \log_2 r \rceil, \quad (5.2)$$

где r – ранг комбинационной схемы.

Однако поскольку сложность комбинационных схем Φ и Ψ значительно больше дополнительных схем для диагностики неисправностей, то дополнительные комбинационные схемы на быстродействие конечного автомата не влияют, а быстродействие конечного автомата определяется комбинационными схемами Φ и Ψ (для конечного автомата Мура наиболее сложной является комбинационная схема Φ). Кроме того, введение дополнительного выходного регистра R_O в ряде случаев приводит даже к увеличению быстродействия конечного автомата. Например, для нашего примера увеличение быстродействия наблюдается для структур VS, TVO и VO (табл. 5).

6. Примеры практического использования рассматриваемых структурных моделей. Выбор структурных моделей конечных автоматов и их объединение выполняется разработчиком устройства управления на основе их характеристик. Следующие факторы должны быть приняты во внимание при выборе структурных моделей:

- неисправности, которые обнаруживаются моделью;
- аппаратные издержки, вычисленные, согласно (5.1);
- быстродействия дополнительных комбинационных схем, определяемые согласно (5.2);
- допустимость задержки выходных сигналов на один такт синхронизации и др.

Рассмотрим выбор наиболее подходящей структурной модели для определения неисправностей на конкретных примерах. В табл. 10 приведены свойства предложенных структурных моделей, которые могут быть приняты во внимание при выборе наиболее подходящей структурной модели конечного автомата Мура из нашего примера.

Например, пусть для нашего примера конечного автомата Мура необходимо выбрать структурную модель, которая позволяет обнаружить наибольшее количество неисправностей и не допускается задержка выходных сигналов на один такт синхронизации, при этом желательно минимизировать увеличение площади и снижение быстродействия конечного автомата. Поскольку для конечного автомата не допускается задержка выходных сигналов на один такт синхронизации, согласно табл. 10, выбирать следует из структур TVI, VI, VNS и VT. Структуру TVI исключаем, поскольку она определяет только одну неисправность (недопустимый входной вектор X). Структуры VNS и VT позволяют выявить наибольшее число неисправностей (табл. 4). Структура

VNS не увеличивает площадь и не снижает быстродействие (табл. 5) по сравнению с базовой структурой, поэтому для реализации выбираем структуру VNS. В результате решаем задачу, не увеличивая площадь и не снижая быстродействие конечного автомата.

При выборе структуры для реализации следует также учитывать количество и сложность обнаруживаемых неисправностей. Пусть, например, для нашего примера конечного автомата Мура поставлена та же задача, но допускается задержка выходных сигналов на один такт синхронизации. Согласно табл. 4, все возможные причины неисправностей можно обнаружить путем объединения любых моделей из двух групп структур: {VNS и VT} и {TVO и VO}. Из группы {VNS и VT} выбираем структуру VNS, которая по сравнению со структурой VT не увеличивает площадь и не снижает быстродействие (табл. 5). Из группы {TVO и VO} выбираем структуру VO, которая позволяет обнаружить более сложные неисправности, чем структура TVO. Поэтому задача решается объединением структур VNS и VO.

Пусть необходимо наиболее точно определить следующие причины неисправностей: недопустимый входной вектор X , сбой в логике Φ , сбой в логике Ψ или сбой в регистре состояний R . Недопустимый входной вектор X определяется с помощью структуры TVI или VI (табл. 4); выбираем структуру VI, которая обнаруживает более сложные неисправности (недопустимые входные векторы в каждом состоянии). Сбой в логике Φ можно определить с помощью структуры VNS или VT; выбираем структуру VNS, которая не увеличивает площадь и не снижает быстродействие. Сбой в логике Ψ можно определить с помощью структуры TVO или VO; выбираем структуру VO, которая обнаруживает более сложные неисправности (недопустимые выходные векторы в каждом состоянии). Сбой в регистре состояний R можно обнаружить с помощью любой структуры, за исключением структуры TVI; выбираем структуру VS, которая однозначно указывает на сбой в регистре R или в цепи обратной связи. Поэтому задача решается объединением следующих структур: VI, VNS, VO и VS.

Пусть необходимо обнаружить переходы в недопустимые состояния, а также недопустимые переходы в допустимые состояния. Данная задача решается с помощью структуры VT.

Если необходимо наиболее точно указать местоположение и причину неисправности (например, при реализации конечного автомата в ASIC), то для реализации следует объединить все структуры, а диагностические сигналы tv_i , vi , vs , vns , vt , tvo и vo вывести на внешние выходы конечного автомата.

Заключение. Представлены структурные модели конечных автоматов Мура, которые позволяют обнаружить следующие неисправности:

- недопустимые входные векторы для всего автомата;
- недопустимые входные векторы в каждом состоянии;
- недопустимый код настоящего состояния;
- недопустимый код следующего состояния;
- недопустимые переходы между состояниями;
- недопустимые выходные векторы для всего автомата;
- недопустимые выходные векторы, формируемые в каждом состоянии.

Приведена также обобщенная структурная модель конечного автомата, позволяющая обнаружить все перечисленные выше неисправности. Кроме того, рассмотренные структурные модели позволяют обнаружить многократные сбои в логике Φ формирования кода следующего состояния, в логике формирования значений выходных сигналов Ψ , в регистре состояний R , а также в цепи обратной связи. Предлагаемые структурные модели конечных автоматов не только обнаруживают неисправности, но и предотвращают их негативное воздействие на управляемый объект.

Будущие исследования будут направлены на разработку структурных моделей, которые допускают исправление сбоев конечного автомата.

СПИСОК ЛИТЕРАТУРЫ

1. Park S., Kim H.T., Lee S., Joo H., Kim H. Survey on Anti-Drone Systems: Components, Designs, and Challenges // IEEE Access. 2021. № 9. P. 42635–42659.
2. Соловьев В.В. Использование методики ASMD-FSMD при проектировании на программируемых логических интегральных схемах устройств обработки сигналов // РЭ. 2021. Т. 66. № 12. С. 1178–1188.

3. *Salauyou V., Zabrocki L.* Coding Techniques in Verilog for Finite State Machine Designs in FPGA // Proc. Intern. Conf. on Computer Information Systems and Industrial Management. Belgrade. Serbia: Springer. Cham, 2019. P. 493–505.
4. *Lyons R.E., Vanderkulk W.* The Use of Triple-Modular Redundancy to Improve Computer Reliability // IBM J. Research and Development. 1962. V. 6. № 2. P. 200–209.
5. *Aviziens A.* Fault-Tolerant Systems // IEEE Transactions on Computers. 1976. V. 100. № 12. P. 1304–1312.
6. *Rochet R., Leveugle R., Saucier G.* Analysis and Comparison of Fault Tolerant FSM Architecture Based on SEC Codes // Proc. IEEE Intern. Workshop on Defect and Fault Tolerance in VLSI Systems. Venice. Italy: IEEE, 1993. P. 9–16.
7. *Niranjan S., Frenzel J.F.* A Comparison of Fault-Tolerant State Machine Architectures for Space-Borne Electronics // IEEE Transactions on Reliability. 1996. V. 45. № 1. P. 109–113.
8. *Carmichael C.* Triple Module Redundancy Design Techniques for Virtex FPGAs // Xilinx Application Note XAPP197.1. 2001.
9. *Pontarelli S., Cardarilli G.C., Malvoni A., Ottavi M., Re M., Salsano A.* System-on-Chip Oriented Fault-Tolerant Sequential Systems Implementation Methodology // Proc. IEEE Intern. Sympos. on Defect and Fault Tolerance in VLSI Systems. San Francisco. USA: IEEE, 2001. P. 455–460.
10. *Lima F., Carro L., Reis R.* Designing Fault Tolerant Systems into SRAM-Based FPGAs // Proc. 40th Annual Design Automation Conf. Anaheim. USA: Machinery, 2003. P. 650–655.
11. *Burke G.R., Taft S.* Fault Tolerant State Machines. Washington. USA: NASA, 2004.
12. *Berg M.A.* Simplified Approach to Fault Tolerant State Machine Design for Single Event Upsets // Mentor Graphics Users' Group User2User Conf. Santa Clara. USA, 2004.
13. *Tiwari A., Tomko K.A.* Enhanced Reliability of Finite-State Machines in FPGA Through Efficient Fault Detection and Correction // IEEE Transactions on Reliability. 2005. V. 54. № 3. P. 459–467.
14. *Cassel M., Lima F.* Evaluating One-Hot Encoding Finite State Machines for SEU Reliability in SRAM-based FPGAs // Proc. 12th IEEE Intern. On-Line Testing Sympos. Lake Como. Italia: IEEE, 2006. P. 139–144.
15. *Frigerio L., Salice F.* RAM-Based Fault Tolerant State Machines for FPGAs // Proc. 22nd IEEE Intern. Sympos. on Defect and Fault-Tolerance in VLSI Systems. Rome. Italy: IEEE, 2007. P. 312–320.
16. *Azambuja J.R., Sousa F., Rosa L., Kastensmidt F.L.* Evaluating Large Grain TMR and Selective Partial Reconfiguration for Soft Error Mitigation in SRAM-Based FPGAs // Proc. 15th IEEE Intern. On-Line Testing Sympos. Lisbon. Portugal: IEEE, 2009. P. 101–106.
17. *El-Maleh A.H., Al-Qahtani A.S.* A Finite State Machine Based Fault Tolerance Technique for Sequential Circuits // Microelectronics Reliability. 2014. V. 54. № 3. P. 654–661.
18. *Sooraj S., Manasy M., Bhakthavatchalu R.* Fault Tolerant FSM on FPGA Using SEC-DED Code Algorithm // Proc. Intern. Conf. on Technological Advancements in Power and Energy. Kollam. India: IEEE, 2017. P. 1–6.
19. *Nidhin T.S., Bhattacharyya A., Behera R.P., Jayanthi T., Velusamy K.* Verification of Fault Tolerant Techniques in Finite State Machines Using Simulation Based Fault Injection Targeted at FPGAs for SEU Mitigation // Proc. 4th Intern. Conf. on Electronics and Communication Systems. Coimbatore. India: IEEE, 2017. P. 153–157.
20. *Choi S., Park J., Yoo H.* Area-Efficient Fault Tolerant Design for Finite State Machines // Proc. Intern. Conf. on Electronics, Information, and Communication. Barcelona. Spain: IEEE, 2020. P. 1–2.
21. *Verducci O., Oliveira D.L., Batista G.* Fault-Tolerant Finite State Machine Quasi Delay Insensitive in Commercial FPGA Devices // Proc. IEEE 13th Latin America Sympos. on Circuits and System. Santiago. Chile: IEEE, 2022. P. 1–4.
22. *Климович А.С., Соловьев В.В.* Структурные модели конечных автоматов при их реализации на программируемых логических интегральных схемах и системах на кристалле // Изв. РАН. ТиСУ. 2015. № 2. С. 68–80.
23. *Yang S.* Logic Synthesis and Optimization Benchmarks user Guide. Version 3.0 // Microelectronics Center of North Carolina (MCNC). North Carolina. USA: MCNC, 1991.
24. *Климович А.С., Соловьев В.В.* Преобразование автомата типа Мили в автомат типа Мура путем расщепления внутренних состояний // Изв. РАН. ТиСУ. 2010. № 6. С. 70–79.
25. *Соловьев В.В.* Синтез быстрых конечных автоматов на программируемых логических интегральных схемах путем расщепления внутренних состояний // Изв. РАН. ТиСУ. 2022. № 3. С. 69–80.