УДК 681.3

М.А. МИРОШНИК, д-р техн. наук, Э.Н. КУЛАК, канд. техн. наук,
ЕГАНА МОВСУМ КЫЗЫ АЛИЕВА, канд. техн. наук, Д.Г. КАРАМАН, Ю.В. ПАХОМОВ

МЕТОДЫ ПРОЕКТИРОВАНИЯ САМОПРОВЕРЯЕМЫХ ЦИФРОВЫХ АВТОМАТОВ

Введение

Для аппаратной реализации цифровых систем и их компонентов все чаще используются программируемые логические интегральные схемы типа CPLD и FPGA ввиду легкости их освоения и относительно невысоких затрат, связанных с отладкой и организацией мелкосерийного производства. По этим причинам большинство разрабатываемых методов синтеза цифровых устройств ориентировано на эту элементную базу, хотя явно это может и не указываться.

С ростом сложности проектируемых и реализуемых решений растет и необходимость обеспечения надежности их функционирования на приемлемом уровне. Для эффективного решения данной проблемы необходимо еще на этапе проектирования заложить возможности проведения технической диагностики. Это достигается правильным выбором элементной базы, рациональным размещением функциональных узлов и блоков, организацией режимов работы и тестовой и функциональной диагностики. Обеспечение таких характеристик как достоверность результатов, минимальное время обнаружения и локализация отказов, возлагается на схемы контроля.

Каждый метод диагностирования требует применения принципиально разных подходов к выбору типа используемых средств и построению диагностической инфраструктуры. При функциональном диагностировании применяются особые методики проектирования и реализации подконтрольных модулей, так называемое тестопригодное проектирование. При построении многоуровневых цифровых систем внедряется дополнительная инфраструктура для того, чтобы результаты диагностирования каждого низкоуровневого модуля могли быть доведены до самого высокого уровня. При тестовом диагностировании создается отдельная инфраструктура, которая включает в себя средства генерации тестовых векторов и средства анализа выходных реакций диагностируемого модуля. В состав средств анализа выходных реакций часто входят схемы свертки множества выходных последовательностей диагностируемого модуля в формат, удобный для анализа исправности модуля, а также для хранения и передачи системам диагностики и мониторинга более высокого уровня. В состав ДИ также входят вспомогательные средства, которые позволяют отключать проверяемое устройство от функционального тракта, и подключать его к основным элементам ДИ.

Надежность и отказоустойчивость функционирования компьютерных систем (КС) и их подсистем относится к стратегическому уровню защиты безопасности систем автоматического и автоматизированного управления. Поддержка работоспособности и исправности функционирования всех программно-аппаратных средств обеспечивается встроенными средствами тестового, функционального диагностирования и восстановления работоспособности, которые включены в систему технического диагностирования (СТД) или диагностической инфраструктуры (ДИ).

Постановка задачи

Самопроверяемые цифровые автоматы определяются как класс полностью самопроверяемых цифровых устройств, надежно защищенных от ошибок определенного типа. Свойство самопроверяемости является обобщенным для всех дискретных объектов, в которых осуществляется оперативное обнаружение ошибок или неисправностей в процессе функционирования объекта, и заключается в покрытии заданного класса неисправностей как в схемах функционального тракта проверяемого устройства, так и в самих средствах

диагностической инфраструктуры. Ошибки и неисправности могут быть устойчивыми или перемежаю-щимися. В технической диагностике математической моделью ошибок и неисправностей являются логические и константные неисправности. Для систем функционального диагностирования дискретных объектов адекватной является модель логических неисправ-ностей, позволяющая учитывать воздействие перемежающихся неисправностей и кратковре-менных сбоев.

Самопроверяемые схемы и устройства широко используются в транспортных, аэро-космических, медицинских системах и т.п. с целью повышения надежности и отказоустой-чивости этих систем, обеспечения уровня их самопроверяемости. Функциональный блок (ФБ) в таких устройствах конструктивно сопряжен со схемой встроенного контроля (СВК), которая проверяет принадлежность выходного слова ФБ заданному множеству разрешенных комбинаций выбранного кода, обнаруживающего ошибки.

В современных цифровых устройствах используется большое количество интегральных микросхем сверхбольшой степени интеграции. К ним можно отнести микроконтроллеры, микропроцессоры, цифровые процессоры обработки сигналов и программируемые логические интегральные схемы (ПЛИС). ПЛИС это универсальный базис для проектирования цифровых устройств любого уровня сложности, который в настоящее время содержит встроенную память, блоки умножения, умножители частоты и прочие встроенные блоки. Спектр применения ПЛИС разнообразен. Они активно используются в аппаратуре специального
назначения, таких, как изделия, применяемые в области авионики, космонавтики, управления ответственными промышленными и транспортными объектами, например, железнодорожной автоматике и телемеханике. В таких устройствах большое внимание уделяется
надежности элементной базы. В том числе остро стоит проблема повышения отказоустойчивости программируемых интегральных схем. Однако следует отметить, что производители микросхем недостаточно развивают эти направления. Несмотря на рост интереса к рынку отказоустойчивых ПЛИС, число их пользователей все еще не слишком велико и производи-тели не видят коммерческой выгоды в проектировании отказоустойчивых кристаллов.

Цель статьи – разработка алгоритмов и методов повышения отказоустойчивости цифровых систем, реализуемых на ПЛИС, путем создания универсальной диагностической инфраструктуры для самопроверяемых цифровых автоматов в базисе логических ячеек ПЛИС.

Анализ состояния вопроса

Вопрос организации диагностической инфраструктуры в аппаратных реализациях систем криптографической защиты данных в общем и в отдельных реализациях алгоритмов шифрования в частности был поднят, после того как начали появляться различные решения аппаратных реализаций алгоритмов шифрования на ПЛИС. Так, в работе [1] авторы рассматривают последствия, к которым приводят возникающие в аппаратных реализациях алгоритмов шифрования ошибки и неисправности, изучают возможность применения двух методов обнаружения ошибок, один из которых основан на введении избыточности, а второй –
на использовании кодов, обнаруживающих ошибки. В работе [2] рассмотрена архитектура построения встроенной системы функционального самодиагностирования для аппаратных реализаций симметричного алгоритма шифрования AES. Предлагаемое решение предназначено для параллельно реализованной архитектуры алгоритма, поскольку активно использует присущие для такой архитектуры циклические повторения одинаковых блоков. Особое внимание в статье уделено тестопригодной реализации подстановочных блоков алгоритма,

поскольку они являются основными нелинейными преобразованиями в алгоритме и требуют для своей реализации максимальное количество ресурсов целевой платформы.

Ряд работ посвящен рассмотрению возможности реализации систем встроенного самодиагностирования для защиты от нового класса криптоаналитических атак, направленных на недостатки в реализации алгоритмов шифрования и эксплуатацию особенностей целевой платформы для взлома криптографических систем [3 – 5].

При аппаратной реализации криптографических алгоритмов, составляющие их преобразования могут быть представлены в виде соответствующих булевых функций. Такой подход к их реализации позволяет с высокой степенью оптимальности адаптировать реализацию криптографического алгоритма на ПЛИС, применять известные методы оптимизации, обеспечивать функциональное и тестовое диагностирование [6, 7].

Развитие субмикронных электронных технологий и широкое использование систем на одном кристалле (SoC), ПЛИС типа FPGA и CPLD, функционирующих на тактовых частотах 1 – 5 ГГц позволяет создавать вычислительные системы и цифровые устройства высокой производительности и быстродействия. С другой стороны в системах и устройствах промышленной автоматики, программируемых логических контроллерах промышленного применения, устройствах локомотивной сигнализации входные воздействия формируются датчиками, преобразователями информации, быстродействие которых в $10^3 – 10^6$ раз меньше быстродействия современных СБИС. Это стало предпосылкой появления и развития систем технического диагностирования (СТД) с интеллектуальными свойствами для проверки исправности и работоспособности дискретных объектов (ДО) путем совмещения процедур их функционального и тестового диагностирования в оперативном режиме. СТД осуществляет диспетчеризацию процессов нормального функционирования ДО и оперативного тестового диагностирования, которое осуществляется в интервалах времени неизменных состояний входных сигналов. В соответствии с предлагаемым подходом процедура проектирования строго безопасного ДО состоит из следующих шагов:

1) проектирование полностью самопроверяемой (ПСП) схемы функциональных блоков (ФБ) ДО и ПСП схемы встроенного контроля ДО, осуществляющей проверку исправности ДО в процессе нормального функционирования на рабочих воздействий;

2) проектирование встроенных средств тестового диагностирования ДО, включающих генераторы тестовых последовательностей и многоканальные синдромно-сигнатурные анализаторы, обеспечивающие проверку исправности ДО для расширенного класса логических и константных неисправностей;

3) проектирование схемы управления и диспетчеризации процедур функционального и тестового диагностирования, восстановления работоспособности и строгой безопасности ДО.

Модель самопроверяемого дискретного автомата

В опубликованных ранее работах [8, 9] разрабатывались различные процедуры синтеза полностью проверяемых цифровых автоматов (ППЦА), комбинационных и последовательностных схем, а также ППЦА схем контроля в соответствии с критериями минимальных

затрат; максимального быстродействия с минимальным числом каскадов, блоков, синтезом легкотестируемых структур и т.п.

В системах автоматики [8] традиционно использовался подход защиты от неисправностей, основанный на дублировании входных и выходных сигналов, обнаружении неисправных узлов с последующей заменой их на резервные экземпляры и восстановлением работоспособности системы. Замена дублированного сигнала кодом 1-из-2 не представляет большой сложности, а многолетний опыт проектирования ППЦА с использованием k-из-2k, или m-из-n кодов может быть использован для создания самопроверяемых цифровых автоматов, модель которого представлена на рис. 1 в виде структуры самопроверяемого цифрового

автомата Мура, где R_1 и R_2 – входной и выходной регистры соответственно; $СВК_1$, $СВК_2$, $СВК_3$ – полностью самопроверяемые схемы встроенного контроля.

В соответствии с требованиями международного стандарта проектирования цифровых устройств IEEE 1149 “Standard Test Access Port and Boundary-Scan Architecture” входные и выходные переменные в процессе выполнения тестового цикла фиксируются в регистрах R_1 и R_2. Каждая входная переменная x_i представляется двухпроводным кодом, в котором «0» и «1» – значения x_i – кодируются векторами («01», «10»). При этом любая однонаправленная ошибка преобразует входную переменную в («00», «11»), что фиксируется в ППЦА схемой контроля $СВК_1$.

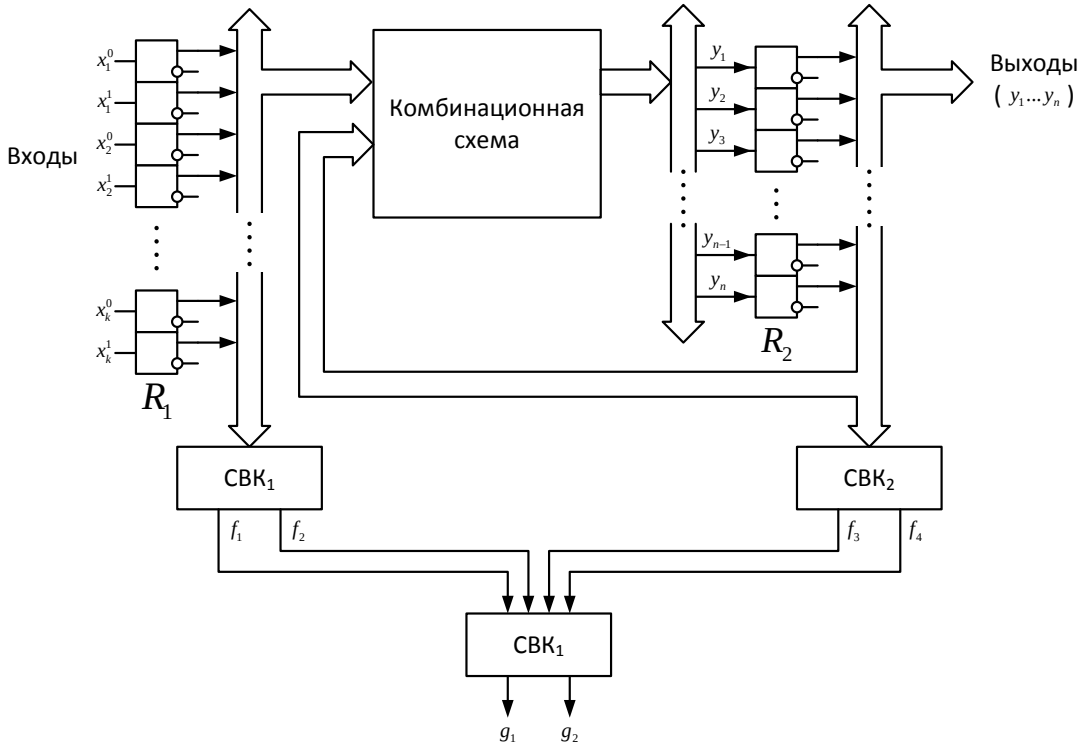

Рис. 1. Структурная модель самопроверяемого цифрового автомата Мура

Выходы автомата Мура, фиксируемые в регистр R_2, кодируются соответствующим неразделимым кодом m-из-n, а преобразование в некодовое слово в результате воздействия неисправностей комбинационной схемы автомата и регистра R_2 обнаруживается в ППЦА схемой контроля $СВК_2$. Двухбитовые выходные линии $СВК_1$ и $СВК_2$ преобразуют код 2-из-4 в двухпроводный код 1-из-2 ППЦА схемой контроля $СВК_3$, выходы которой g_1 и g_2 являются индикатором исправности цифрового автомата.

Свойство самопроверяемости цифровых автоматов

Пусть задан автомат $M = (X,Z,Y,\delta,\lambda)$, где $X = \{x_1, x_2,\ldots,x_k\}$ – множество входных переменных, образующих k-мерное входное векторное пространство из 2^k векторов; $Z = \{z_1,z_2,\ldots,z_q\}$ – множество состояний автомата; $Y = \{y_1,y_2,\ldots,y_n\}$ – множество выходных переменных, образующих выходное n-мерное векторное пространство из 2^n векторов; δ, λ – функции переходов и выходов автомата соответственно. Если автомат M исправен, то в соответствии с определениями 1, 2 и 3, приведенными в [8], в процессе функционирования на его входы поступает только подмножество векторов $X_a \subseteq X$, которое называется входным кодовым пространством, а на выходах формируется подмножество $Y_b \subseteq Y$, которое
называется выходным кодовым пространством. Элементы кодовых пространств называют

кодовыми словами. При появлении неисправностей (ошибок) формируются некодовые слова. Эффективность схемы ППЦА основана на трех фундаментальных предположениях:

1) схема является самопроверяемой и защищенной от ошибок для любой одиночной неисправности $r \in \Phi$;

2) устойчивая неисправность $r_1 \in \Phi$ не изменяется на протяжении времени поступления на входы всех кодовых слов;

3) неисправность порождает однонаправленные ошибки в выходном кодовом пространстве;

Однако второе предположение не всегда выполняется в устройствах промышленного применения. Во-первых, в процессе функционирования таких устройств зачастую не все подмножество X_a входных кодовых слов поступает на входы устройства в определенных интервалах функционирования, а значит, неисправность r_1 может быть не обнаружена на выходе схемы. Во-вторых, в этом интервале функционирования может появиться неисправность r_2 такая, что $r_1 \cup r_2 \notin \Phi$. Тогда нарушается сделанное ранее предположение.

Пусть множество неисправностей $\Phi=\{r_1,r_2,\ldots,r_n\}$ появляются в схеме в произвольной последовательности в процессе функционирования устройства.

Схема является самопроверяемой, если для любой последовательности неисправностей $[r_1,r_2,\ldots,r_n]$ из множества неисправностей Ф на выходе ее формируется кодовое или некодовое слово на входном векторном пространстве, не обязательно совпадающим с пространством $X_a \subseteq X$ входных кодовых слов.

Корректность определения подтверждается результатами исследований защищенных от расширенного класса константных неисправностей схем ППЦА.

Таким образом, самопроверяемые схемы цифровых автоматов являются расширением схем ППЦА, проверка исправности которых в процессе функционирования должна осуществляться множеством входных векторов, обнаруживающих последовательности одиночных неисправностей, появляющихся на определенных интервалах функционирования устройств.

В [8] предложен метод проектирования самопроверяех схем, основанный на анализе структуры ФБ, множества самопроверяемых путей транспортировки сигналов на выходы схемы и последующей модификации структуры схемы для обеспечения ее полной самопроверяемости в соответствии с определениями.

Проектирование самопроверяемых автоматов

Представленный в [8] в качестве примера цифровой автомат выбора кода (АВК) автоматической локомотивной сигнализации на входе имеет семь функциональных сигналов $(x_1,x_2,\ldots,x_7)$ и два управляющих сигнала (RST, CLK). В результате анализа основных функций АВК по контролю наличия или отсутствия условий, необходимых для выбора требуемого кодового сигнала в пределах контролируемого блок-участка, контроля ситуации на

защитном участке, а также анализа сигнальных ситуаций была разработана модель цифрового автомата Мура, которая представлена графом переходов автомата в указанной работе [8, рис. 3].

В качестве анализатора реакций на тестовые последовательности можно использовать типовые схемы многоканальных сигнатурных анализаторов. Предложенное решение позволяет исключить трудоемкие процедуры генерации детерминированных проверяющих тестов и моделирование неисправностей.

Проверка исправности АВК исчерпывающей тестовой последовательности позволяет обнаруживать класс логических и кратных константных неисправностей в соответствии с условиями самопроверяемости АВК.

Проектирование схемы управления и диспетчеризации процедур функционального и тестового функционирования АВК можно выполнить путем использования стандартных и известных методов синтеза управляющих автоматов.

Одной из важнейших частей диагностической инфраструктуры является система генерации и подачи тестовых воздействий на диагностируемый объект. Существует множество вариантов такой системы, среди которых наиболее очевидным является организация массива памяти, хранящего все тестовые векторы. Однако если даже для каждого проверяемого объекта выделять память на сотни проверяющих тестов, то в совокупности такая диагностическая инфраструктура может потребовать объемы памяти для тестов, в несколько раз превышающих доступные ресурсы даже наиболее производительных аппаратных платформ (кристаллов ПЛИС).

Существует еще один общеизвестный метод, при котором затраты на построение системы генерации тестовых воздействий являются минимальными. Его суть заключается в подаче на диагностируемый объект полного множества возможных входных воздействий, которые генерируются с помощью счетчиковых структур, сдвиговых регистров с линейной обратной связью и клеточных автоматов. Главным достоинством такого подхода является исключение дорогостоящих и трудоемких процедур генерации проверяющих тестов и моделирования неисправностей, а также снижение затрат на итоговую схему генератора [9].

Для рассмотренной ранее схемы автомата одним из наиболее приемлемых с точки зрения аппаратных затрат вариантом является встроенный генератор тестовых последовательностей, который реализован на семиразрядном сдвиговом регистре с нелинейной обратной связью (СРНОС), описываемой булевой функцией $f_{oc} = x_7 \oplus \left(x_1 + \overline{x_2 + x_3 + x_4 + x_5 + x_6} \right)$. Генераторы на СРНОС получили значительно меньшее распространение, чем генераторы на базе сдвиговых регистров с линейными обратными связями из-за того, что функциональную зависимость обратной связи отыскать намного сложнее, и методы их описания не вписываются в удобную математическую модель полиномиальной арифметики над конечным полем GF(2^n). Однако генераторы на СРНОС обладают одним преимуществом над генераторами на базе СРЛОС: они способны выдавать полное множество выходных последовательностей 2^n, тогда как генератор на СРЛОС выдает неполную последовательность $2^n - 1$. Такая последовательность не включает тривиальный тест, битовая проекция которого состоит из логических 0, тогда как этот тестовый вектор присутствует практически в любом наборе тестов.

Реализация СРНОС на языке VHDL приведена на рис. 2.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity test_vector_gen_7bit is
         port(
           CLK : in STD_LOGIC;
           CEN : in STD_LOGIC;
           RST : in STD_LOGIC;
           TG_OUTP : out STD_LOGIC_VECTOR(6 downto 0)
         );
end test_vector_gen_7bit;

architecture tg_7bit_arc of test_vector_gen_7bit is
        signal tg_buffer : std_logic_vector(6 downto 0);
        signal feedback_f : std_logic;
begin

        -- register
        SR: process(CLK, RST)
        begin
          if (RST='1') then
                tg_buffer <= (others => '0');
```

```
            elsif (CLK'event and CLK = '1') then
                  if (CEN = '1') then
                        tg_buffer <= tg_buffer(5 downto 0) & feedback_f;
                  end if;
            end if;
          end process;

          -- combinatorial part: feedback function and output assignment
          feedback_f   <=   tg_buffer(6)   XOR   (tg_buffer(0)   OR   NOT(tg_buffer(1)   OR
tg_buffer(2) OR tg_buffer(3) OR tg_buffer(4) OR tg_buffer(5)));
          TG_OUTP <= tg_buffer;

    end tg_7bit_arc;
```

Рис. 2. Листинг кода СРНОС на языке VHDL

Результаты верификации спроектированной схемы приведены на рис. 3. Верификация работы генератора осуществлялась с помощью трех сигналов: на вход CLK был подан периодический сигнал тактовой рабочей частоты с периодом 100 нс (соответствует частоте 10 МГц), на вход RST подан сигнал логической «1» в течение первых двух тактовых циклов, чтобы обеспечить гарантированный сброс верифицируемой схемы. Уровень логической «1», поданный на вход CEN, выполняет запуск перебора последовательности тестовых векторов. Как видно из рисунка, зацикливание перебираемой последовательности произошло на 13, 85 мкс от начала моделирования. При условии, что начало процесса генерации приходится на 1050 нс, имеем 13850 – 1050 = 12800 или 128 тактовых циклов по 100 нс. Таким образом видно, что генератор обеспечивает полный исчерпывающий перебор бесповторной
последовательности всех 2^7 = 128 тестовых векторов за не более чем 13 мкс работы системы при тактовой частоте 10 МГц.

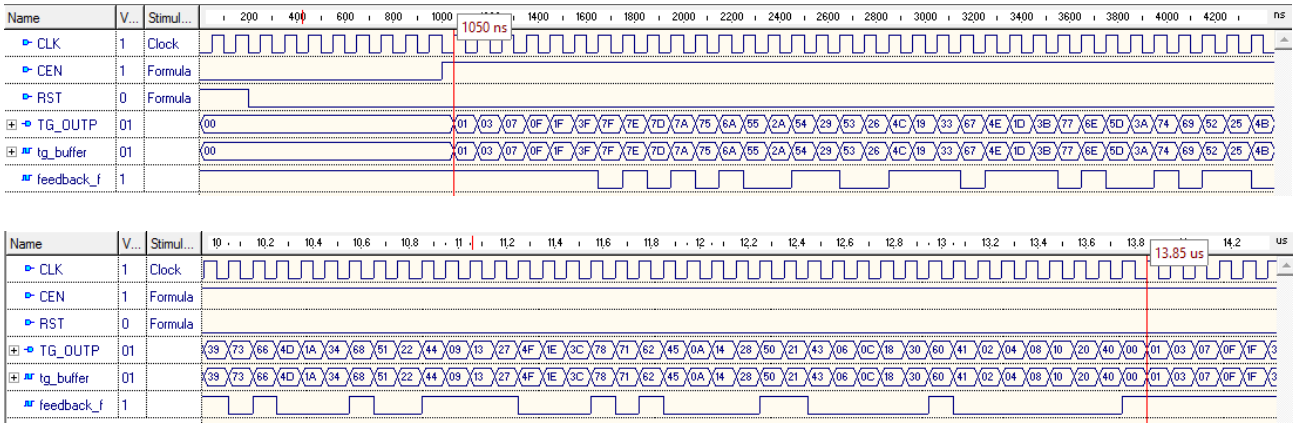

Рис. 3. Результаты верификации СРНОС

Как видно из рисунка, генератор позволяет получить все возможные входные комбинации ($128 = 2^7$).

На рис. 4 представлены результаты синтеза генератора тестов при выборе в качестве целевой платформы микросхемы ПЛИС xc3s100e из семейства Spartan-3E фирмы Xilinx. Реализация схемы в базисе ПЛИС фирмы Xilinx занимает 2 4-входовых LUT-элемента и 7 триггеров, что в сумме составляет 4 Slice-ячейки.

test_vector_gen_7bit Project Status			
Project File:	tg_7bit.xise	**Parser Errors:**	No Errors
Module Name:	test_vector_gen_7bit	**Implementation State:**	Synthesized
Target Device:	xc3s100e-5tq144	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	4	960	0%
Number of Slice Flip Flops	7	1920	0%
Number of 4 input LUTs	2	1920	0%
Number of bonded IOBs	10	108	9%
Number of GCLKs	1	24	4%

Рис. 4. Результаты синтеза генератора тестов на базе СРНОС

Затраты на АВК, представленный в работе [8], составили 13 4-входовых LUT-элементов и 4 триггера (7 Slice-ячеек). Если рассматривать процентное соотношение ресурсов, затрачиваемых на реализацию проверяемого автомата и диагностической инфраструктуры, то предлагаемое решение в виде генератора проверяющих тестов на базе СРНОС составляет около 57 % от затрат на АВК по количеству Slice-ячеек.

Заключение

В результате проведенных исследований введено и обосновано понятие класса самопроверяемых цифровых устройств, разработана автоматная модель, обеспечивающая свойства самопроверяемости. Предложен метод проектирования самопроверяемых автоматов,
основанный на совмещении процедур функционального и тестового диагностирования, использовании методов синтеза полностью самопроверяемых схем и синтеза схем встроенного тестового контроля с использованием методов компактного тестирования. Разработанный метод проектирования самопроверяемых автоматов, основан на совмещении процедур функционального и тестового диагностирования, использовании методов синтеза полностью самопроверяемых схем и синтеза схем встроенного тестового контроля с использованием методов компактного тестирования.

Предложенный метод является универсальным и может быть использован как при проектировании цифровых систем управления, так и для разработки надежных модулей криптографической защиты данных. Применение предложенных моделей и методов демонстрируется на примере проектирования самопроверяемого автомата. На примере автомата АВК, приведенного в работе [8], была синтезирована модель генератора проверяющих тестов на базе генератора СРНОС для диагностической инфраструктуры полностью самопроверяемого цифрового автомата, которая показала степень затрат на уровне 57 % от затрат на реализацию самого цифрового автомата. Данное приращение аппаратных затрат является несущественным, т.к. суммарно автомат с его диагностической инфраструктурой занимает меньше 1 % площади кристалла ПЛИС xc3s100e семейства Spartan-3E фирмы Xilinx.

Список литературы: 1. Bertoni, G. et al. Error analysis and detection procedures for a hardware implementation of the advanced encryption standard // IEEE Transactions on Computers. – 2003. – Т. 52. – №. 4. – P. 492-505. 2. Di Natale, G., Flottes, M. L., Rouzeyre, B. On-Line Self-Test of AES Hardware Implementations // The 37-th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007. 3. Blömer, J., Krummel, V. Analysis of countermeasures against access driven cache attacks on AES //International Workshop on Selected Areas in Cryptography. – Springer Berlin Heidelberg, 2007. – P. 96-109. 4. Opritoiu, F., Bozesan, A., Vladutiu, M. Pseudo random self-test architecture for Advanced Encryption Standard // Design and Technology in Electronic Packaging (SIITME), 2013 IEEE 19th International Symposium for // IEEE. – 2013. – P. 271-276. 5. Mathew, J. et al. On the synthesis of attack tolerant crypto-

graphic hardware // 2010 18th IEEE/IFIP International Conference on VLSI and System-on-Chip // IEEE. – 2010. – С. 286-291. 6. Дербунович, Л. В., Караман, Д. Г., Методы функционального диагностирования ошибок шифрования в симметричных криптографических системах // Вестник НТУ «ХПИ». – №57. – С. 81-86. 7. Дербунович, Л. В., Караман, Д. Г., Пащенко, Т. Н. Метод синтеза древовидных легкотестируемых логических схем // Вестник НТУ "ХПИ". Автоматика и приборостроение. – 2009. – Вып. 23. – С. 64-70. 8. Мирошник, М. А. Метод проектирования строго безопасных автоматов локомотивной сигнализации. / Дербунович Л. В., Малиновский М. Л., Караман Д. Г., Мирошник М. А. Осипенко А. Н. // Інформаційно-керуючі системи на залізничному транспорті. – 2012. – №5. – С. 25-42. 9. Кулак, Э.Н., Ларченко, Л.В., Филиппенко, И.В. Метод анализа тестопригодности цифровых схем при генерации взвешенного псевдослучайного теста // Науч.-техн. и практ. журнал. – Уральск : ТОО «Уралнаучкнига», 2014. – № 42 (121). – С 70 -78.

Украинский государственный университет железнодорожного транспорта, г. Харьков;
Харьковский национальный университет радиоэлектроники;
Азербайджанский государственный университет нефти и промышленности, г. Баку, Азербайджан
Харьковский национальный университет «ХПИ»
Харьковский национальный университет городского хозяйства имени А.Н. Бекетова,

Поступила в редколлегию 31.10.2016