УДК 517.2+519.2

ТЕСТОВОЕ ДИАГНОСТИРОВАНИЕ АППАРАТНОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

В.Н. ЯРМОЛИК, А.А. ИВАНЮК

Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220027, Беларусь

Поступила в редакцию 8 января 2014

Кратко изложены основные научные и практические результаты исследований в области тестового диагностирования вычислительных систем, полученные в БГУИР в рамках представляемой научной школы. Приведены основные характеристики оригинальных решений в области стендового оборудования для тестирования цифровых модулей, контролепригодного синтеза вычислительных систем, методов компактного тестирования, теории сигнатурного анализа и методов самотестирования вычислительных машин и систем. Описаны новые оригинальные авторские методы псевдослучайного и вероятностного тестирования, исчерпывающего, псевдоисчерпывающего и почти исчерпывающего тестирования, управляемого случайного тестирования и оптимального управляемого случайного тестирования, а также квазислучайного тестирования программного обеспечения. Представлены результаты посвященные тестированию, самотестированию и саморемонтированию запоминающих устройств, а также неразрушающему тестированию оперативных запоминающих устройств (ОЗУ) с применением адаптивного сигнатурного анализа и симметричных маршевых тестов ОЗУ. Кроме того, приведены основные результаты по методам обеспечения целостности данных с использованием цифровых водяных знаков и запутывающих преобразований и их применению по обеспечению авторского права на программное обеспечение. Представлены результаты, полученные в области физической криптографии.

Ключевые слова: тестовые последовательности, псевдослучайные последовательности, квазислучайные последовательности, сигнатурный анализ, адаптивный сигнатурный анализ, вероятностное тестирование.

Введение

Научное направление, название которого отражено в заглавии статьи, сформировалось в БГУИР (ранее, МРТИ) в 80-ые гг. как одно из направлений широко известной научной школы профессора Леусенко А.Е. и при всесторонней поддержке, на протяжении многих лет, ректора МРТИ Ильина В.М.

Зарождение данного направления было связано с активным развитием проектирования и производства в СССР электронных вычислительных машин (ЭВМ) серии ЕС. Большая номенклатура машин данного семейства, применение интегральных схем малой и средней интеграции и большие объемы выпускаемых средств вычислительных техники поставили новые задачи технической диагностики по созданию средств тестового диагностирования для серийно выпускаемых ЭВМ серии ЕС. Дальнейшее развитие данного научного направления связано с большим циклом исследований в области компактного тестирования и его основных компонент, а именно псевдослучайного тестирования и сигнатурного анализа как основных элементов систем встроенного тестирования вычислительных систем. Большое количество работ научной школы связано с тестированием и самотестированием запоминающих устройств в части развития теории маршевых тестов ОЗУ и неразрушающего тестирования и диагностирования на базе адаптивного сигнатурного анализа и симметричных маршевых

Библиотека БГУИР

тестов. Дальнейшие исследования посвящены разработке методов тестирования современных вычислительных систем, таких как встроенные системы (Embedded System), системы на кристалле (System-on-a-Chip) и сети на кристалле (Net-on-a-Chip). В настоящее время, в рамках научной школы, проводятся исследования по разработке универсальных методов по тестированию вычислительных систем, а также по разработке методов обеспечения целостности данных с использованием достижений современной стеганографии (Steganography), физической криптографии (Physical Cryptography) и теории запутывающих преобразований (Obfuscation). Показана эффективность применения разработанных авторами систем водяных знаков (Watermarking) и оттисков пальцев (Fingerprinting) для обеспечения авторского права на программное обеспечение и проекты аппаратных средств, представленных на языке VHDL. Активно развивается теория и практика физически неклонируемых функций (PUF) как радикального средства для идентификации и аутентификации современных сверхбольших интегральных схем (СБИС).

В исследованиях по различным аспектам указанного научного направления принимали участие студенты, аспиранты, магистранты и сотрудники кафедр ЭВМ и ПОИТ БГУИР (МРТИ). Многие из приведенных результатов получены совместно с зарубежными исследователями и учеными из: Лондонского университета (University of London, UK); Бостонского университета (Boston University, USA); Стэнфордского университета (Stanford University, USA); Гренобльского университета (Institute National Polytechnigue de Grenoble, France); Делфтского технологического университета (Delft University of Technology, Denmark); Штутгардского университета (University of Stuttgart, Germany); Вупертальского университета (Wuppertal University, Germany); Наньянского технологического университета (Nanyang Technological University, Singapore) и Политехнического университета Белостока (Politechnika Bialostocka, Poland).

Результатом проведенных исследований явилась защита четырех докторских диссертаций, более 20 кандидатских диссертаций, большого количества магистерских диссертаций и дипломных проектов.

Подробное описание решений задач тестового диагностирования вычислительных систем, предложенных авторами, можно найти в более чем 500 публикациях, включающих 17 монографий, изданных на трех языках, и 72 авторских свидетельства на изобретение. Наиболее значимые публикации авторов, включая монографии и статьи, приведены в качестве библиографии к данной статье.

Стендовое оборудование для изделий сложной техники

Зарождение в МРТИ научного направления, связанного с технической диагностикой сложных технических систем, связано с именем известного ученого А.Е. Леусенко. Он стоял у истоков становления МРТИ как высшего учебного заведения СССР. Под его руководством была развита теория и практика виброиспытаний изделий сложной техники. Одним из авторов данной статьи, учеником А.Е. Леусенко, была предложена идея псевдослучайного диагностирования [1, 2]. Преимущества псевдослучайного диагностирования, заключающиеся в обеспечении стабильных характеристик входных тестовых воздействий и практически нулевой дисперсии получаемых оценок, позволили применить его для проведения виброиспытаний изделий сложной техники [3, 4]. Разработанные и внедренные «Цифровой имитатор широкополосных случайных процессов» и «Имитатор для генерирования псевдослучайных процессов», по сравнению с известными, на то время, устройствами аналогичного класса, позволяли осуществлять программное управление точностью воспроизведения спектральной плотности мощности. Процесс управления в подобном стендовом оборудовании заметно сокращается за счет использования псевдослучайных последовательностей с укороченным периодом [3–6].

Становление научного направления было связано с активным развитием проектирования и производства в СССР электронных вычислительных машин серии ЕС. Большая номенклатура машин данного семейства, применение интегральных схем малой и средней интеграции и большие объемы выпускаемых средств вычислительной техники сформулировали новые задачи технической диагностики по созданию сервисного стендового

Библиотека БГУИР

оборудования для тестового диагностирования серийно выпускаемых ЭВМ серии ЕС. Применение принципов псевдослучайного и вероятностного диагностирования позволило эффективно диагностировать типовые элементы замены (ТЭЗ) и модули серийно выпускаемых ЭВМ, таких как: ЕС-1036, ЕС-1061, ЕС-1130, ЕС-1842 и др. [7–13]. Для этих целей было разработано и внедрено сервисное стендовое оборудование КДФК, СКАТ и ДУКАТ, основным режимом функционирования, которых является режим компактного тестирования с использованием псевдослучайных тестовых воздействий [14–19]. В сравнении с существовавшими на то время разработками, предназначенными для формирования тестовых воздействий, КДФК, СКАТ и ДУКАТ отличились высокой достоверностью диагностирования за счет эффективности псевдослучайных тестов и высокой достоверностью сигнатурного анализа [20, 21]. Результаты экспериментальных исследований и длительного промышленного использования стендов КДФК, СКАТ и ДУКАТ на заводе ЭВМ им. Г.К. Орджоникидзе, Брестском производственном объединении средств вычислительной техники (БПО СВТ) и п/я М-5339 подтвердили высокое качество разработанных методов псевдослучайного тестирования, что позволило рекомендовать расширение сферы их применения в практике производства изделий вычислительной техники [7–22].

Разработанный пакет прикладных программ СИНПОЛ позволяет синтезировать генераторы псевдослучайных исчерпывающих тестов для СБИС со структурой сквозного сдвигового регистра и цифровых модулей, использующих стандарт JTAG (IEEE 1149.1). Высокая эффективность пакета СИНПОЛ и генераторов псевдослучайных исчерпывающих тестов, синтезированных с его использованием, позволила применить его для реализации тестирования всех 108 типовых элементов замены ЭВМ ЕС-1130 [23, 24]. Так, для ТЭЗ АЛУ1 ЕС-1130 в результате синтеза был получен примитивный порождающий полином $\varphi(x)=1\oplus x\oplus x^3\oplus x^4\oplus x^6\oplus x^9\oplus x^{10}\oplus x^{11}\oplus x^{16}\oplus x^{21}$, а для ТЭЗов Управление АЛУ и ЛП полином $\varphi(x)=1\oplus x\oplus x^2\oplus x^6\oplus x^9\oplus x^{10}$, как основа для построения генераторов псевдоисчерпывающих тестов [24].

Компактное тестирование

Развита теория построения тестовых воздействий для современной элементной базы, цифровых устройств, регулярных структур, СБИС и систем на их основе [9, 13, 14, 20, 25, 26]. Доказана эффективность применения методов случайного поиска для генерирования тестов, основанных на применении псевдослучайных последовательностей [13, 20, 26]. Развита теория *M*-последовательностей как источника псевдослучайных воздействий в части эффективного формирования сдвинутых копий *M*-последовательностей, определения величины сдвига, вычисления коэффициентов связи, уточнения и развития свойства децимаций и теоретико-числового метода определения символов *M*-последовательностей [13, 14, 20, 26]. Доказана эффективность применения *M*-последовательностей для формирования источников псевдослучайных и вероятностных тестовых сигналов с заданными статистическими характеристиками для эффективного псевдослучайного диагностирования цифровых устройств и систем, а также виброиспытаний изделий сложной техники [20, 26]. Предложены методики синтеза генераторов тестовых воздействий для функциональных модулей и *ТЭЗ*ов вычислительных машин [13, 20, 26].

Обоснована необходимость использования методов сжатия выходных реакций при тестировании и самотестировании цифровых устройств и систем [26, 27]. Развита теория сигнатурного анализа [26−35]. Впервые определено понятие сигнатуры $S = S(\varphi(z),A_0,\phi(z),X_1,L,F)$, как функции зависящей от многих переменных, где $\varphi(z)$ и $\phi(z)$ – примитивные полиномы, используемые для построения генератора тестов и сигнатурного анализатора; A_0 – значение начального состояния сигнатурного анализатора; X_1 – начальный тестовый набор тестовой последовательности; L – длина теста; $F=F(x_1,\ldots,x_m)$ булева функция, описывающая исследуемое цифровое устройство [18−20, 26−32]. При фиксированных параметрах $\varphi(z)$, A_0, $\phi(z)$, X_1 и L, введено понятие сигнатуры булевой функции как $S = S(F)$, используя которое определена сигнатурная тестируемость, позволяющая сформулировать правила контролепригодного синтеза цифровых устройств и систем [26]. Для случая комбинационных цифровых устройств понятие сигнатурной тестируемости формулируется следующей теоремой [26].

Библиотека БГУИР

Теорема 1. *Условием сигнатурной тестируемости для одиночной константной неисправности* $\sigma \in \{\equiv 0, \equiv 1\}$, *возникшей на полюсе, описываемом функцией* $g = g(x_1,\ldots,x_m)$, *в комбинационной схеме, реализующей булеву функцию* $F = F(x_1,\ldots,x_m) = gG_1 + \bar{g}\,G_2 + G_3$; $G_1 = G_1(x_1,\ldots,x_m)$, $G_2 = G_2(x_1,\ldots,x_m)$ *и* $G_3 = G_3(x_1,\ldots,x_m)$, *а* $G_1 + G_2 \neq 0$ *и* $G_3 \neq 1$, *является выполнение следующей системы неравенств*: $S(gG_1\overline{G_3}) \oplus S(gG_2\overline{G_3}) \neq 0$, $S(\bar{g}\,G_1\overline{G_3}) \oplus S(\bar{g}\,G_2\overline{G_3}) \neq 0$.

Предложены методики оценки эффективности сигнатурного анализа для встроенного самотестирования устройств и систем. Получены аналитические выражения для меры эффективности сигнатурного анализа как вероятности P_n^{μ} необнаружения μ-кратной ошибки, определяемые следующей теоремой [26, 35].

Теорема 2. *Вероятность* P_n^{μ} *необнаружения сигнатурным анализатором* μ*-кратной ошибки определяется как*:

$$P_n^1 = P_n^2 = 0; \quad P_n^{\mu} = P_n^{\mu+1} = \frac{1}{2^m - \mu}\left(1 - \mu \times P_n^{\mu-2}\right), \quad \mu = 2k+1, \quad k = \overline{1, 2^{m-1} - 2}.$$

Показана инвариантность вероятности P_n^{μ} относительно сжимаемой последовательности и от вида примитивного полинома φ(z). Доказано, что единственным параметром, влияющим на значение P_n^{μ}, является старшая степень $m = \deg\varphi(z)$ полинома φ(z) [35].

Контролепригодное проектирование

Использование принципов псевдослучайного диагностирования позволило создать эффективные средства для реализации тестирования и самотестирования микро-ЭВМ, БИС и СБИС и характеризуется большими функциональными возможностями в сравнении с известными решениями, такими как, например метод BILBO [30, 35]. Предложенные средства самотестирования имеют высокую достоверность и требуют для своей реализации небольших аппаратурных затрат.

Разработана методология проектирования самотестируемых и саморемонтируемых цифровых устройств и систем, основанная на методах контролепригодного синтеза и теории компактного тестирования [35−38]. Обоснованы требования к средствам самотестирования СБИС и предложены методики для их синтеза. Сформулирована методика построения самотестируемых цифровых систем, основанная на применении универсальных многофункциональных тестовых модулей и принципов псевдоисчерпывающего тестирования. Дано определение *k*-псевдоисчерпывающих тестов [39].

Определение 1. *k-псевдоисчерпывающим тестом является матрица* $B(N,k,w)$ *строками которой являются вектора* $B_i = (B_i^{(0)}, B_i^{(1)},\ldots,B_i^{(N-1)})$, $B_i^{(j)} \in GF(2^w)$; $i = 0,1,\ldots,T_k-1$; $j = 0, 1,\ldots, N-1$, *такие, что в матрице*

$$\boldsymbol{B(N,k,w)} = \begin{vmatrix} B_0^{(0)} & B_0^{(1)} & B_0^{(2)} & \ldots & B_0^{(N-1)} \\ B_1^{(0)} & B_1^{(1)} & B_1^{(2)} & \ldots & B_1^{(N-1)} \\ B_2^{(0)} & B_2^{(1)} & B_2^{(2)} & \ldots & B_2^{(N-1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ B_{Tk-1}^{(0)} & B_{Tk-1}^{(1)} & B_{Tk-1}^{(2)} & \ldots & B_{Tk-1}^{(N-1)} \end{vmatrix},$$

все q^k *q-ичные вектора* $(y_0, y_1, y_2,\ldots, y_{k-1},)$ *состоящие из k q-ичных цифр* $(q=2^w)$ *будут сформированы, по крайней мере, один раз в любых k столбцах матрицы.*

Развита теория и практика псевдоисчерпывающих тестовых последовательностей, основанная на следующей теореме и ее следствиях [35, 39].

Теорема 3. *Для* $q = 2^w$, *примитивного элемента* α *над* $GF(q)$ $(\alpha^l \neq \alpha^j;\ l \neq j;$ $l, j = 0, 1, 2,\ldots,q-2)$, *примитивного элемента* β *над* $GF(q^k)$ $(\beta^l \neq \beta^j;\ l \neq j;\ l, j = 0, 1, 2,\ldots, q^k-2)$, *и матрицы*

Библиотека БГУИР

$$H = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha^{1} & \alpha^{2} & \dots & \alpha^{q-2} \\ 1 & \alpha^{2} & \alpha^{4} & \dots & \alpha^{2(q-2)} \\ 1 & \alpha^{3} & \alpha^{6} & \dots & \alpha^{3(q-2)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \alpha^{k-1} & \alpha^{2(k-1)} & \dots & \alpha^{(k-1)(q-2)} \end{vmatrix},$$

обозначив $\beta^{i-1}=(\alpha^{i0},\alpha^{i1},\dots,\alpha^{i(k-1)})\in GF(q^k)$ и $B_0=(0,0,\dots,0)$, $B_i=(B_i^{(0)},B_i^{(1)},\dots,B_i^{(q-1)}) = (\alpha^{i0},\alpha^{i1},\dots,\alpha^{i(k-1)})H$, *где* $B_i^{(j)}\in GF(q)$, $i=1,2,\dots, q^k$ *получим, что*:

1) *для любых* $j_0\le j_1\le\dots\le j_{k-1}$ *и любых* $A_{j0}, A_{j1}, \dots, A_{j(k-1)} \in GF(q)$, *существует такое* $i\in(0, 1, \dots, q^k-1)$ *для которого выполняется система равенств* $B_i^{(j0)} = A_{j0}$, $B_i^{(j1)} = A_{j1}$, …, $B_i^{(j(k-1))} = A_{j(k-1)}$;

2) *для любых* $s\in(0,1,\dots, q-2)$, $j_0\le j_1\le\dots\le j_{k-3}$ $(s \notin (j_0,j_1,\dots,j_{k-3}))$ *и для любых* $A_{j0}, A_{j1}, \dots, A_{j(k-3)}, A_s, A_s'\in GF(q)$, *исключая случай, когда* $A_{j0}=A_{j1}=\dots=A_{j(k-3)}=A_s=A_s'=0$, *существует такое* $i\in(0,1,2,\dots,q^k)$ *для которого выполняется система равенств* $B_i^{(j0)} = A_{j0}$, $B_i^{(j1)} = A_{j1}$, …, $B_i^{(j(k-3))} = A_{j(k-3)}$, $B_i^{(j(k-1))} = A_s$, $B_i^{(jk)} = A_s'$.

Примерами *k*-псевдоисчерпывающих тестов являются $B(4,2,1)$=(0000, 0111, 1011, 1110) и $B(3,2,2)$=(000, 013, 022, 031, 103, 112, 121, 130, 202, 211, 220, 233, 301, 310, 323, 332)

Тестирование запоминающих устройств

Большой цикл работ авторов посвящен тестированию запоминающих устройств (*ЗУ*) [40−48]. В области маршевых тестов основное внимание уделялось разработке тестов, обнаруживающих сложные неисправности, в которых участвуют две и более запоминающих ячейки [49, 50]. Авторами был предложен новый маршевый тест March M, способный обнаруживать связанные неисправности следующих конфигураций: *TF-CF*, *CFid-CFid, CFin-CFid*, *CFin-CFin* (в конфигурации участвует нечетное количество *CFin*) [50]. Тест состоит из восьми маршевых элементов, которые имеют следующий вид:

$\{\Updownarrow(w0); \Downarrow(r0, w1, r1, w0); \Updownarrow(r0); \Downarrow(r0, w1); \Updownarrow(r1); \Downarrow(r1, w0, r0, w1); \Updownarrow(r1); \Uparrow(r1, w0)\}$

Тест March M является минимальным маршевым тестом, обнаруживающим все множество обнаруживаемых неисправностей запоминающих устройств (ЗУ), в которых участвует не более чем две ячейки ЗУ. Развивая теорию неразрушающего тестирования ЗУ, был предложен эффективный маршевый тест минимальной сложности [45]. Он состоит из двух маршевых элементов, и его сложность оценивается как 3*N*. Тест имеет следующий вид: $\{\Uparrow(ra, w\bar{a}); \Updownarrow(r\bar{a})\}$ [45].

Весьма плодотворной оказалась идея многократных маршевых тестов для ОЗУ [51, 62, 72]. Авторами было установлено, что многократные маршевые тесты, имеющие линейную зависимость временной сложности от *N* емкости ОЗУ, позволяют достичь высокой обнаруживающей способности сложных неисправностей, таких как кодочувствительные неисправности PNPSFk, в которых участвует *k* ячеек ОЗУ [70]. В рамках многократного тестирования авторами был предложен метод определения оптимальных сочетаний начальных состояний ОЗУ, основанный на выборе двоичных векторов состояний ОЗУ с максимальным минимальным расстоянием Хэмминга между любой парой начальных состояний. Доказано, что для двукратного MATS+ подобного теста максимальное минимальное расстояние Хэмминга не должно быть меньше чем *N*–*k*, а для трехкратного и четырехкратного применения маршевых тестов ОЗУ не может быть больше чем 2*N*/3. Использование оптимальных сочетаний начальных состояний ОЗУ позволяет повысить эффективность обнаружения сложных кодочувствительных неисправностей. Так в случае четырехкратных MATS+ подобных тестов полнота покрытия PNPSF3 увеличивается с 43,75 %, полученных для известных решений, до 47, 25 % [70].

Разработан метод многократного тестирования, основанный на изменении начальных адресов в адресных последовательностях при повторном применении маршевого теста ОЗУ, основанный на максимизации разностей начальных адресов в последовательных тестовых

Библиотека БГУИР

процедурах. Оптимальный выбор начальных адресов в случае трехкратного MATS++ подобного теста позволяет увеличить полноту покрытия PNPSF5 с 14,00 % до 17,70 %, а для March C – с 24,87 % до 32,90 % [70].

Доказана эффективность использования арифметического расстояния для выбора оптимальных сочетаний модификаций адресов ОЗУ. Так, в случае теста MATS++ и оптимальных циклических сдвигов битовых последовательностей адресов полнота покрытия PNPSF3 увеличивается с 28,0 % до 48,0 % [66, 70].

Предложена методика многократного тестирования ОЗУ, основанная на применении различных алгоритмов формирования адресных последовательностей, и методики оценки эффективного их сочетания, основанные на применении численных характеристик, которые используют расстояние Хэмминга. Показано, что для двукратного тестирования оптимальным сочетанием адресных последовательностей является счетчиковая адресная последовательность и последовательность анти-Грея, что позволяет достичь 43,0 % полноты покрытия PNPSF3 двукратным тестом MATS++ [68−70, 72−74]

Самотестирование и саморемонтирование запоминающих устройств

Развитием основ сигнатурной тестируемости явилась разработка авторами теории и практики адаптивного сигнатурного анализа (Adaptive Signature Analysis (*ASA*)), эффективно используемого для целей самотестирования и саморемонтирования ОЗУ. Основой теории ASA является следующая теорема [58, 70, 75].

Теорема 4. *Сигнатура S произвольного содержимого бит-ориентированного ОЗУ, с* $2^m - 1$ *ячейками, равна* $S = s_m s_{m-1} s_{m-2} ... s_3 s_2 s_1$, *где* $s_i = x_{1,i} \oplus x_{2,i} \oplus x_{3,i} \oplus ... \oplus x_{r,i}$; $x_{ij} \in \{0,1\}$, $i = 1,2,3,..., m$, $j = 1,2,3,..., r$ и $x_{j,1}$, $x_{j,2}$, $x_{j,3}$, …, $x_{j,m}$ *есть адреса ячеек ЗУ, содержащих* 1, *а r – равняется количеству таких ячеек ОЗУ, когда: генератор тестовых последовательностей* (*ГТП*) *и сигнатурный анализатор* (*СА*) *описываются взаимообратными примитивными полиномами* $\varphi(x)$ и $\varphi^{-1}(x)$ *для которых* $\deg\varphi(x) = \deg\varphi^{-1}(x)$; *начальные состояния ГТП и СА равны* (1,0,0,…0) *и* (0,0,0,…,0), *соответственно; число псевдослучайных тестовых наборов, подаваемых на тестируемое ОЗУ, равно* $2^m - 1$ *и включает все двоичные комбинации из m бит кроме нулевой.*

Эффективность *ASA* базируется на следующих его свойствах [56].

Свойство 1. *Сигнатуры для произвольного содержимого ОЗУ и для инверсного содержимого равны между собой.*

Свойство 2. *Конечное значение сигнатуры не зависит от вида адресной последовательности и направления перебора адресного пространства ОЗУ при вычислении сигнатуры.*

Свойство 3. *Все однократные ошибки обнаруживаются и диагностируются. При этом адрес ячейки с ошибочным значением вычисляется как* $A_E = S_{ref} \oplus S_{test}$, где S_{ref} *есть значение эталонной сигнатуры, а* S_{test} *– значение рабочей сигнатуры.*

Свойства 4. *Все двукратные ошибки обнаруживаются, а ошибки большей кратности обнаруживаются с вероятностью* $1–1/2^m$.

Анализ маршевых тестов, проведенный авторами, показал наличие в них четырех видов симметрии данных, что явилось основой разработки семейства симметричных неразрушающих маршевых тестов (Symmetric Transparent Algorithm (STA)) и их применения для самотестирования и саморемонтирования ОЗУ [52, 53, 70, 81]. Основой STA является метод формирования сигнатур независимых от сжимаемых данных, для случая симметрии Type 2 данных $A\,\overline{A^{-1}}$, где $\overline{A^{-1}}$ является инверсной бинарной последовательностью формируемой в обратном порядке по отношению к *A*. Для данных с симметрией Type 2 справедлива следующая теорема [49, 50, 70, 81].

Теорема 5. *Если при сжатии последовательности данных A на сигнатурном анализаторе с начальным состоянием* S_0, *описываемом полиномом* $\varphi(x)$, *была получена сигнатура* $S=S(A,S_0,\varphi(x))$, *тогда сигнатура S, получаемая при сжатии последовательности данных* A^{-1} *на сигнатурном анализаторе, описываемом полиномом* $\varphi(x)^{-1}$, *с начальным*

Библиотека БГУИР

состоянием S^{-1} *будет равняться обратному первоначальному состоянию* S_0^{-1}, *т.е.* $S = S(A^{-1},S(A,S_0,\varphi(x))^{-1}, \varphi(x)^{-1}) = S_0^{-1}$.

Основные свойства неразрушающих тестов построенных на основе *STA* представим в виде утверждений [49, 50, 52, 53, 63, 64, 65, 67, 70, 81].

Утверждение 1. *Процедура STA тестирования является неразрушающей.*

Утверждение 2. *Эталонная сигнатура* S_{ref}, *также как и реальная сигнатура* S_{real} *при наличии неисправностей в ОЗУ, полученные в соответствии с STA, имеют нулевое значение* $S_{ref} = S_{real} = 000\ldots0$ *и не зависят от: размера тестируемого ОЗУ и его содержимого; используемого теста ОЗУ имеющего симметрию Type 2; вида примитивного порождающего полинома* $\varphi(x)$.

Утверждение 3. *Покрывающая способность неразрушающих тестов не ниже покрывающей способности классических неразрушающих тестов.*

Более высокая покрывающая способность может быть получена в результате добавления к тесту ОЗУ дополнительных операций чтения для обеспечения симметрии Type 2.

Утверждение 4. *Реализация STA тестирования требует минимальных аппаратурных затрат, выражающихся в затратах только на реализацию сигнатурного анализатора.*

Самотестируемые цифровые устройства с низким потреблением энергии

Показано, что реализация самотестирования цифровых устройств значительно снижает стоимость и повышает процент покрытия неисправностей, так как проверка может производиться на рабочих частотах, и не требуется внешнего стендового оборудования [55, 56, 61, 82]. Тем не менее, применение самотестирования значительно увеличивает рассеиваемую мощность при тестировании цифровых устройств. По сравнению с обычным режимом функционирования, эта мощность может увеличиваться в 2–3 раза, что приводит к значительному повышению температуры кристалла. Известно, что эффективность тестирования прямо пропорциональна переключательной активности тестируемой схемы, что приводит к значительному увеличению переключательной активности в тестовом режиме по сравнению с нормальным режимом работы.

Для минимизации потребляемой энергии при самотестировании СБИС авторами предложен ряд оригинальных подходов, основанных на:

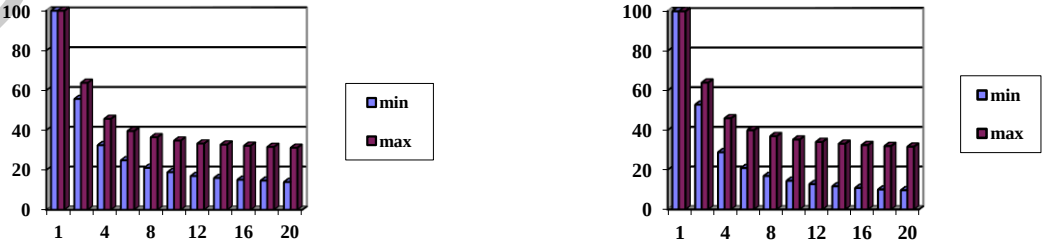
– структурных методах, использующих различные архитектуры самотестирования СБИС типа test-per-clock;

– методах минимизации переключательной активности;

– специальных методиках для архитектур самотестирования СБИС типа test-per-scan;

– методах синтеза средств самотестирования, таких как генератор тестов и анализатор выходных реакций [55, 56, 61, 82].

Для иллюстрации эффективности предлагаемых решений по снижению потребляемой энергии на рис. 1 приведены результаты синтеза генераторов тестов. За основу взяты две структуры классических генераторов тестов соответствующих полиномам со старшими степенями $m = 10$ и $m = 20$ [56]. Принимая переключательную активность классических генераторов за 100 %, были получены оценки для предлагаемых решений в зависимости от количества d ($d = 2,\ldots,20$), одновременно формирующих бит тестовой последовательности за один такт синхронизации.



а *б*

Рис. 1. Переключательная активность ГПТН при различных значениях *d: m*=10 (*а*); *m* = 20 (*б*)

Библиотека БГУИР

Анализ графиков показывает, что в среднем при d = 2 экономится порядка 40 % энергии, при d = 4 – уже 60 %. Дальнейшее увеличение d до 20 приводит к экономии еще 20 %. Для практического применения целесообразно использование d = 2,3,4 для которых достигается компромисс между уменьшением энергоемкости генератора тестов и его аппаратурной сложности.

Управляемое вероятностное тестирование программного обеспечения

Большое внимание, в рамках представляемой научной школы, уделяется вероятностному тестированию (Random Testing) цифровых устройств и программного обеспечения, и методологии черного ящика (Black Box) его применения. Авторами показано, что различные модификации антивероятностного тестирования такие как: быстрое антивероятностное тестирование (Fast Antirandom (FAR)); адаптивное вероятностное тестирование (Adaptive Random Testing); эволюционное вероятностное тестирование (Evolutionary Random Testing); эффективное вероятностное тестирование (Good Random Testing); ограниченное вероятностное тестирование (Restricted Random Testing); зеркальное вероятностное тестирование (Mirror Random Testing); упорядоченное вероятностное тестирование (Orderly Random Testing) и другие решения, основаны на вычислении характеристик для потенциального кандидата в тесты на основании предыдущих тестовых наборов, и, соответственно, характеризуются большой вычислительной сложностью [76−78].

Предложенный авторами метод генерирования оптимальных управляемых случайных тестов (Optimal Controlled Random Tests) (OCRT) является обобщением известных алгоритмов использующих в своей основе «жадный алгоритм», а также расстояние Хемминга и декартово расстояние в качестве критериев выбора очередного тестового набора [78, 86]. В отличие от известных решений предложенный метод гарантированно обеспечивает максимально возможные значения расстояний Хемминга $HD(T)$ и Декарта $CD(T)$. Оптимальные управляемые OCRT имеют очевидное преимущество по отношению ко всем известным методам формирования управляемых случайных тестов [86]. Они характеризуются минимальной вычислительной сложностью в силу отсутствия процедур перечисления всевозможных кандидатов в тестовые наборы и вычисления для них соответствующих характеристик. Для произвольного N наиболее трудоемкой процедурой при формировании OCRT является процедура построения матрицы M состоящей из $q = 2(m+1)$ строк и выполнения перестановок ее столбцов для получения уникальной матрицы масок.

Управляемые вероятностные тесты OCRT, по сравнению с известными решениями, позволяют повысить эффективность формирования всевозможных двоичных комбинаций на любых r из N разрядах генерируемых тестовых наборов. Так, для случая оптимальных управляемых случайных тестов, принимая допущение, что $N^2>>N$ и $r<<N$, можно показать, что $WE(OCRT,r)$ – взвешенное процентное отношение r-разрядных двоичных комбинаций, формируемых $OCRT$, будет вычисляться в соответствии с выражением [76].

$$WE(OCRT,r)=\left(1-\left(\frac{2^{r-1}-1}{2^{r-1}}\right)^{q/2}\right)100\%;\quad q=2,4,6,...$$

Приведенное соотношение является универсальной метрикой эффективности управляемых вероятностных тестов для случая, когда OCRT состоит из четного числа $q = 2(\lceil \log_2 N \rceil+1)$ наборов и включает последовательные инверсные пары наборов $T_i = \overline{T}_{i-1}$, $i \in \{1,3,5,\ldots,q-1\}$.

Для генерирования оптимальных управляемых вероятностных тестов малой длины (Short Optimal Controlled Random Tests (SOCRT)), с малым количеством наборов q, авторами были рассмотрены коды с максимальным минимальным расстоянием Хемминга ($Max\_minHD(T_i,T_j)$) [84−86]. Теорема Плоткина позволяет определить максимально возможное количество q кодовых слов в двоичном коде длины N для минимального кодового расстояния d, а граница Плоткина дает верхний предел этого количества. Для случая, когда $2d - N > 0$ согласно теореме Плоткина справедливо следующее соотношение $d \le qN/(2(q-1))$ [86].

Библиотека БГУИР

При построении OCRT для $q = 2$, в качестве второго тестового набора T_1 использовалась инверсия первого T_0 набора, т. е. $T_1 = \overline{T}_0$. В этом случае достигается максимальное значение $HD(T_0,T_1)=N$, тогда SOCRT(2,N)={$T_0, \overline{T}_0$}. Для $q = 3$, Max_min$HD(T_i,T_j)\leq 3N/4$. Авторами доказано, что ближайшее оптимальное решение может быть получено только для $Max\_minHD(T_i,T_j) = 2N/3$, здесь $2N/3<3N/4$. Для $N = 6$, соответствующий SOCRT(3,$2N$/3) = SOCRT(3,4) = {T_0,T_1,T_2} = {000000, 111100, 001111}. Для случая $q = 4$, Max_min$HD(T_i,T_j) \leq 2N/3$. Оптимальный тест *SOCRT*(4,$2N$/3) имеет Max_min$HD(T_i,T_j) = 2N/3$, что соответствует теореме Плоткина. Для $N = 6$ получим, что SOCRT(4,$2N$/3) = SOCRT(4,4) = {T_0,T_1,T_2,T_3} ={000000, 111100, 001111, 110011}. Следует отметить, что приведенные примеры SOCRT(3,4) и SOCRT(4,4), являются не единственными решениями для $N = 6$. Кроме того, для заданного $q = 4$, возможны и другие решения для меньших значений Max_min$HD(T_i,T_j)$, чем $2N/3$, но также близких к оптимальным значениям. Одним из таких решений является тест SOCRT(4,5N/8) с Max_min$HD(T_i,T_j)=5N/8 < 2N/3$. Примерами SOCRT(4,5N/8) могут быть SOCRT(4,5)={T_0,T_1,T_2,T_3}= {00000000, 00011111, 11111000, 11100111} и SOCRT(4,10) = {T_0,T_1,T_2,T_3}={0000000000000000, 00000011 11111111, 1111111111000000, 1111110000111111} для $N = 8$ и 16. Два приведенных примера SOCRT(4,$2N$/3) и SOCRT(4,5N/8), очевидно являются наилучшими решениями, соответствующими условиям теоремы Плоткина. Авторами показано, что подобным образом процедура построения тестов SOCRT(q,d) для малых значений d может быть продолжена, однако в каждом конкретном случае соответствующий тест является эвристическим результатом, требующим большого объема исследований.

Квазислучайное тестирование вычислительных систем

Архитектурные особенности современных вычислительных систем, многообразие физических дефектов их аппаратной части и ошибок в программном обеспечении, а также многочисленные подходы для тестирования таких систем предопределяют необходимость применения универсальных методов для совместного тестирования аппаратной и программной частей систем. Авторами проанализированы основные источники неисправностей вычислительных систем и показаны их причинно-следственные связи, приведенные на рис. 2

Основными недостатками методов вероятностного тестирования являются их невысокая полнота покрытия неисправностей и большая длина тестовых последовательностей. Подобными недостатками характеризуется и метод Монте-Карло, для которого характерна значительная вычислительная погрешность и заметная временная сложность. Поэтому в качестве альтернативного решения для тестирования вычислительных систем авторами предлагается использование идеи квазислучайного тестирования, основанного на применении квазислучайных последовательностей, в качестве тестовых последовательностей, эффективно покрывающих пространство входных воздействий систем.

Последовательность неслучайных чисел называется квазислучайной, если ее можно использовать в реализации алгоритмов Монте-Карло вместо случайной последовательности. Именно такие последовательности используются на практике для различных задач метода КМК, что позволяет достичь меньших вычислительных погрешностей и более быстрой сходимости. Это достигается не столько свойством независимости, характерным для псевдослучайных последовательностей, сколько свойством равномерности. Такие последовательности в русскоязычной литературе называют, согласно Соболю, $ЛП_\tau$-последовательностями, что интерпретируется следующим образом: любой последовательный участок хорошо распределен (более равномерно по сравнению с псевдослучайными последовательностями). В англоязычной литературе такие последовательности называют последовательностями с малым дискрепансом (low-discrepancy sequence), а их разновидности – по именам авторов, выделяя последовательности Соболя.

Основой для формирования модифицированных квазислучайных последовательностей Соболя, предложенных авторами, является порождающая матрица V, состоящая из модифицированных направляющих чисел и представляющая собой нижнюю треугольную матрицу (унитреугольную матрицу) с единичной главной диагональю. Развивая методологию

Библиотека БГУИР

квазислучайного тестирования, был предложен универсальный генератор входных тестовых воздействий, свойства последовательностей которого определяются видом квадратной $m \times m$ матрицы V в соответствии со следующим утверждениями [87, 88].

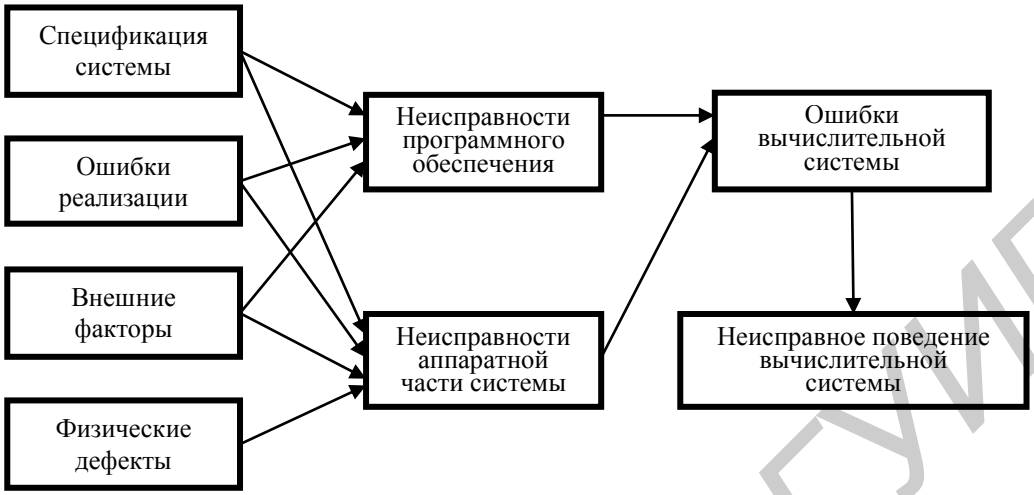


Рис. 2. Связь между неисправностями, ошибками и неисправным поведением вычислительной системы

Утверждение 5. *Модифицированная последовательность Соболя определяется видом порождающей матрицы V, представляющей собой нижнюю треугольную матрицу (унитреугольную матрицу) с единичной главной диагональю.*

***Утверждение** 6. Пересчетная последовательность определяется видом порождающей матрицы V, представляющей собой нижнюю треугольную матрицу относительно побочной единичной диагонали.*

***Утверждение 7.** Порождающая матрица V для тестовой последовательности с минимальной переключательной активностью состоит из m отличающихся друг от друга строк, каждая их которых содержит по одной единице.*

***Утверждение 8.** Порождающая матрица V для адресной последовательности с предельной максимальной переключательной активностью состоит из одного единичного столбца и m – 1, отличающихся друг от друга столбцов, содержащих по одному нулевому значению в каждой из m – 1 строк, кроме первой.*

Основываясь на приведенных утверждениях, оказывается возможным формирование широкого спектра тестовых последовательностей, таких как: пересчетные последовательности; последовательности Корпута (Corput); последовательности Грея (Gray); последовательности анти-Грея; последовательности с заданной переключательной активностью; последовательности с максимально возможной переключательной активностью и др.

Системы водяных знаков и оттисков пальцев

Известно, что количество нелегальных копий программных продуктов ведущих фирм достигает от 30 до 98 % в зависимости от региона и страны. В среднем в мире около 35 % инсталлированного программного обеспечения является пиратским. Выделяют три основные категории атак на современное программное обеспечение: нелегальное использование или воровство ПО (Software Piracy); обратное проектирование (Reverse Engineering) – процесс незаконного извлечения наиболее ценных модулей или частей исходного кода, и несанкционированная модификация кода (Software Tampering), приводящая к существенным потерям, вплоть до потери разработчиками ПО авторских прав. Системы водяных знаков и оттисков пальцев, позволяют обеспечит защиту авторского права на программные продукты.

Определены характеристики программного обеспечения, допускающие их использование для внедрения информации об авторе (владельце): автокорреляционная характеристика машинного кода, соотношение частот использования машинных команд, характеристика числовой последовательности, получаемой в результате интерпретации

Библиотека БГУИР

очередности следования независимых машинных команд [57, 60]. Проведенное исследование методов модификации характеристик программных модулей, скомпилированных под архитектуру IA32, показало, что метод, основанный на перестановках независимых команд, является наиболее эффективным с точки зрения объема скрываемой информации [57, 60].

Разработанный метод внедрения водяного знака в программное обеспечение на уровне исходного текста, основанный на одностороннем преобразовании выражений, стоящих в условных конструкциях, является устойчивым к атаке типа удаление и искажение [57, 60].

Разработаны методы внедрения водяного знака (признака авторства) на уровне машинного кода, основанные на преобразовании машинного кода, модифицирующем вид одной из выделенных статистических характеристик программного модуля, обладающие повышенной устойчивостью к обнаружению и применимые для защиты готовых бинарных исполняемых модулей [57, 60].

Предложена адаптация метода Patchwork, ранее использованного для внедрения водяного знака в растровые изображения, позволяющая осуществлять скрытую модификацию характеристик машинного кода, выполняемую при размещении признака авторства [57, 60].

Предложена схема совместного использования разработанных методов в виде многоуровневой системы защиты, которая обеспечивает повышенную стойкость внедряемой информации и выполняет функцию защиты от обратного проектирования и модификации кода. Предложенная схема реализована в виде программного комплекса, превосходящего лучшие из доступных аналогов по таким характеристикам как: границы применения, защищенность от известных атак и влияние на качество защищаемой программы. Разработанная система практически используется на предприятии для защиты исходных кодов и бинарных исполняемых модулей разрабатываемого программного обеспечения.

Методы запутывающих преобразований

Запутывающие преобразования программных приложений являются эффективным средством для защиты их авторского права и/или правообладания. Пример результата обфускации приведен на рис. 3.

```
#include "stdio.h" #define e 3 #define g (e/e) #define h ((g+e)/2) #define f (e-g-h) #define j
(e*e-g) #define k (j-h) #define l(x) tab2[x]/h #define m(n,a) ((n&(a))==(a)) long
tab1[]={989L,5L,26L,0L,88319L,123L,0L,9367L}; int tab2[]={ 4,6,10,14,22,26,34,38,46,58,
62,74, 82,86 }; main(m1,s) char *s; {int a,b,c,d,o[k],n=(int)s;if(m1==1){ char b[2*j+f-g];
main(l(h+e)+h+e,b); printf(b); }else switch(m1-=h){case f:a=(b=(c=(d=g)<<g)<<g)<<g;
return(m(n,a|c)|m(n,b)|m (n,a|d)|m (n,c|d)); case h:for(a=f;a<j;++a)
if(tab1[a]&&!(tab1[a]%((long)l(n))))return(a);case g:if(n<h)return(g);if(n<j){n-=g;c='D';
o[f]=h;o[g]=f;} else {c='\r'-'\b';n-= j-
g;o[f]=o[g]=g;}if((b=n)>=e)for(b=g<<g;b<n;++b)o[b]=o[b-h]+o[b-g]+c;return(o[b-g]%n+k-
h);default:if(m1-=e) main(m1-g+e+h,s+g); else *(s+g)=f;for(*s=a=f;a<e;)
*s=(*s<<e)|main(h+ a++,(char *)m1);}}
```

Рис. 3. Обфусцированная программа, результатом работы которой является вывод фразы *Hello, world*!

Авторами предложена метрическая модель оценки эффективности запутывающего преобразования, основанная на представлении программы в виде вектора в пространстве измеренных для нее метрик сложности и потребления ресурсов [54, 60, 71]. Очевидно, что наилучшим является запутывающий преобразователь, который оптимизирует код по времени и/или по объему памяти (уменьшает значение модуля вектора $E\downarrow$) и вместе с тем делает его сложнее (увеличивается модуль вектора $E\uparrow$). Поэтому вектор представления программы рассматривается как сумма двух векторов:

$$\vec{p}' = \vec{E}_{\uparrow} + \vec{E}_{\downarrow}.$$

Представленная модель позволяет сравнивать различные стратегии запутывания. Даны оценки эффективности широко используемых методов запутывающего преобразования (лексического метода и метода внедрения в код предикатов). Эти оценки позволяют сделать вывод о целесообразности применения лексического метода и высокой эффективности метода

Библиотека БГУИР

внедрения в код предикатов при его рациональной реализации. Для лексического метода запутывания исследован вопрос о статистическом распределении длин идентификаторов, полученные результаты должны быть использованы при попытке сокрытия факта запутывания.

Разработан метод искусственного увеличения сцепления модулей, что противодействует такой атаке как использование модуля целиком без исследования алгоритмов его работы. Метод основан на внедрении инструкций одних модулей в другие [54, 60]. Кроме того, в рамках рассмотрения этого модуля представлена концепция целостности программных средств, позволяющая оценить целесообразность разделения программ на отдельные библиотеки модулей, связанных функциональной зависимостью [54, 60].

Разработан алгоритм изменения структуры данных программ, основанный на преобразовании максимального числа полей классов и локальных переменных в единый массив базового типа, что позволяет использовать приемы запутывания массивов и увеличивает возможности лексической трансформации [54, 60]. Реализован программный комплекс Obfuscation Studio, преимущества которого заключаются в расширяемости и простоте использования. Данный комплекс практически применяется для защиты программ, написанных для платформы *.NET* от обратного проектирования, незаконного исследования и модификации.

Физическая криптография

Помимо описанных видов атак на программное обеспечение, действие которых распространяется и на проектные HDL-описания, к цифровым устройствам чаще применяется атака типа «клонирование». Методы, нацеленные на предотвращение подобных несанкционированных действий, получили название методов физической криптографии. Одним из актуальных направлений в физической криптографии является исследование и реализация физически неклонируемых функций (PUF – Physically Unclonable Functions). Задача извлечения уникальных параметров цифровых устройств является основой для создания PUF.

Области применения PUF: идентификация и аутентификация цифровых устройств, защита от клонирования, генераторы действительно случайных числовых последовательностей (True Random Number Generation), цифровые водяные знаки и оттиски пальцев (Hardware Watermarking and Fingerprinting), обнаружение и защита от аппаратных модификаций (Hardware Trojan Detection) и т.п. В рамках исследований методов физической криптографии были разработаны новые типы комбинированных Strong PUF: Arbiter&RS-Latch PUF и Arbiter&RO PUF, эффективность которых в сравнении с существующими решениями была доказана экспериментально [79]. Так для Arbiter&RS-Latch PUF в среднем было получено следующее распределение ответов: *Ri* = '0' (39 %), *Ri* = '1' (61 %), при этом для стандартной реализации Arbiter PUF такое распределение выглядит следующим образом: *Ri* = '0' (80 %), *Ri* = '1' (20 %).

Был предложен новый механизм арбитража для PUF типа Arbiter, позволяющий не только производить сравнение задержек распространения тестового импульса по симметричным путям, но и регистрировать изменения скважности тестового импульса, что в совокупности позволяет более эффективно решать задачу индентификации цифровых устройств [83].

Все экспериментальные работы по исследованию свойств PUF цифровых устройств проводились с применением макетных плат быстрого прототипирования Digilent NEXUS-2 с ПЛУ типа FPGA Xilinx SPARTAN XC3S500E, САПР цифровых устройств Xilinx ISE, системы моделирования ModelSim, и набора программных компонент Digilent Adept SDK. Разработана схема PUF конфигурируемого кольцевого генератора, позволяющая идентифицировать различные экземпляры цифровых устройств, реализуемых ПЛУ типа FPGA [80]. На рис. 3 представлена схема конфигурируемого кольцевого генератора, реализованного на FPGA Xilinx SPARTAN XC3S500E.

Аналитически было показано, что различие в числе регистрируемых импульсов для двух реализаций генератора на различных ПЛУ можно оценить по формуле $\Delta N_R(k) \approx k\left\lfloor \frac{\delta D_S}{2D_{RO}^2} \right\rfloor$,

где D_S – продолжительность временного окна измерения числа зарегистрированных импульсов N_R, D_{RO} – значение задержки распространения сигнала в цепи обратной связи генератора,

Библиотека БГУИР

$\delta = D^*_{RO} - D_{RO}$ – величина различия двух сравниваемых генераторов, k – натуральное число, определяющее коэффициент масштабирования D_S.

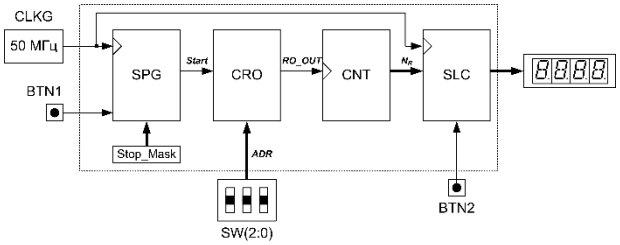


Рис. 4. Структура конфигурируемого кольцевого генератора, реализованного в системе Digilent NEXUS-2

Экспериментально было установлено, что для $k > 2^7$ значение $\Delta N_R(k)$ принимает не нулевые значения для всех возможных ADR (рис. 5).

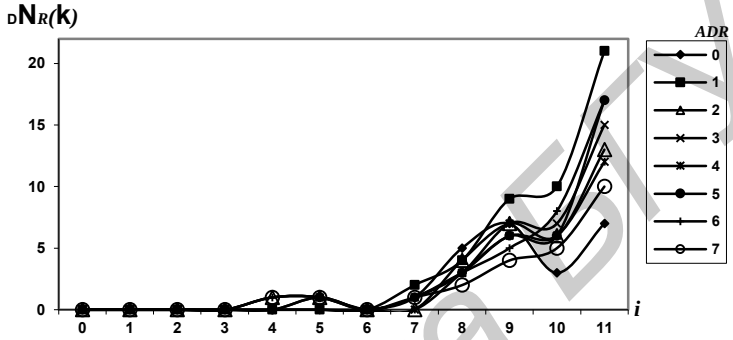


Рис. 5. Значения разницы $\Delta N_R(k)$ для $k = 2^i$ и всех возможных значений ADR

Согласно полученным экспериментальным данным для значений $i \geq 7$ ($i = \lceil \log_2 k \rceil$) наблюдается стабильное увеличение арифметического расстояния $D^*_\Delta(k)$ между числом импульсов, зарегистрированных в процессе моделирования, и числом импульсов, зарегистрированных на реальных цифровых системах. В свою очередь, значение $D^*_\Delta(k)$ определяет число уникальных идентификаторов (число исследуемых микросхем ПЛИС) для выбранного значения k. Например, для $k = 2^{11}$ около 400 ПЛИС из класса XC3s500e-5FG320 могут быть различимы. Таким образом, увеличение значения k позволяет повышать достоверность идентификации различных реализаций цифрового устройства [80].

Разработаны цифровые генераторы действительно случайных числовых последовательностей, основанных на свойствах нестабильности и непредсказуемости схемной реализации модифицированной RO-SRAM PUF [90].

Оценка основных характеристик генерируемых последовательностей производилась посредствам программных пакетов Statistica, NIST и DIEHARD. Вырабатываемые числовые последовательности успешно проходят статистические тесты на соответствие эталонной случайной последовательности. Таким образом, разработанные генераторы обладают свойствами неклонируемости и невоспроизводимости, а вырабатываемые последовательности – свойствами непериодичности, некоррелируемости и равномерности.

Полученные экспериментальные результаты по исследованию свойств различных типов физически неклонируемых функций дают основание для поиска новых схемотехнических решений, позволяющих одновременно осуществлять извлечение стабильных уникальных характеристик интегральных схем и выполнять роль источника энтропии.

Заключение

В лаборатории активно проводятся исследования методов обфускации проектных HDL-описаний и методов схемотехнического запутывания цифровых устройств для затруднения их

Библиотека БГУИР

понимания и обратного проектирования, для внедрения водяных знаков и доказательства авторства. Осуществляется поиск схемотехнических решений Strong PUF для ПЛУ типа FPGA с целью решения задач идентификации. В перспективе научных исследований лаборатории находятся методы активного измерения аппаратуры (Active Hardware Metering) цифровых устройств для выявления фактов их нелегального использования и запрещения функционирования. Планируется объединение накопленного опыта в обеспечении надежного функционирования средств вычислительной техники с методами защиты аппаратных и программных компонент интеллектуальной собственности от несанкционированных изменений и использования. В перспективе – разработка новых методов, алгоритмов и средств проектирования надежных самотестируемых, самодиагностируемых, саморемонтируемых цифровых устройств и систем, обеспечивающих конфиденциальность, достоверность и надежность обрабатываемых данных в рамках методологии Design For Trust.

Полученные научные результаты докладывались на многих отечественных и зарубежных конференциях и симпозиумах, среди которых можно выделить следующие: International Test Conference (ITC), VLSI Test Symposium (VTS), Defect and Fault-Tolerance in VLSI Systems (DFT), European Design and Test Conference (ED&TC), Design, Automation and Test in Europe (DATE), Asian Test Symposium (ATS), Field-Programmable Logic and Applications (FPL), Mixed Design of Integrated Circuits and Systems (MIXDES), Design and Diagnostics of Electronics Circuits and Systems (DDECS), Computer Information Systems and Industrial Management Applications (CISIM), Computer-Aided Design of Discrete Devices (CAD DD), Современные проблемы радиоэлектроники, Технические средства защиты информации, Информационные технологии в промышленности, Информационные технологии и системы.

TEST DIAGNOSTICS OF COMPUTER SYSTEMS HARDWARE AND SOFTWARE

V.N. YARMOLIK, A.A. IVANIUK

Abstract

The main scientific and practical results on test diagnostics of hardware and software of computer systems are presented in brief history. More than 500 international and local publications including 17 books and 72 invention certificates reinforce the scientific results. The current state of research and a plan of future research of laboratory are presented.

Список литературы

1. *Леусенко А.Е., Ярмолик В.Н., Петровский А.А.* // Изв. ВУЗов. Приборостроение. 1977. № 12. С. 39–42.
2. *Ярмолик В.Н., Леусенко А.Е.* // Изв. ВУЗов. Приборостроение. 1979. № 7. С. 3–7.
3. *Леусенко А.Е., Петровский А.А., Ярмолик В.Н.* // Измерительная техника. 1980. № 10. С. 23–26.
4. *Ярмолик В.Н., Леусенко А.Е., Морозевич А.Н.* // Изв. ВУЗов. Приборостроение. 1982. № 12. С. 31–34.
5. *Ярмолик В.Н.* // Изв. ВУЗов. Приборостроение. 1983. № 1. С. 48–52.
6. *Havel J, Yarmolik V.N., Morozevich A.N.* // Kibernetik. 1983. № 1. P. 58–65.
7. *Ярмолик В.Н.* // Электронное моделирование. 1983. № 5. С. 49–55.
8. *Ярмолик В.Н.* // Изв. ВУЗов. Приборостроение. 1983. № 11. С. 80–82.
9. *Ярмолик В.Н.* / АиТ. 1983. № 6. С. 155–162.
10. *Ярмолик В.Н.* // АиТ. 1985. № 1. С. 127–132.
11. *Ярмолик В.Н.* // Электронное моделирование. 1985. № 6. С. 51–54.
12. *Ярмолик В.Н.* // АиВТ. 1985. № 4. С. 73–79.
13. *Ярмолик В.Н., Демиденко С.Н.* Генерирование и применение псевдослучайных сигналов в системах испытаний и контроля. Минск, 1986.
14. *Ярмолик В.Н.* // Радиотехника. 1986. № 6. С. 54–57.
15. *Ярмолик В.Н., Кацнельсон Е.И.* // АиВТ. 1986. № 4. С. 82–86.
16. *Ярмолик В.Н.* // Микроэлектроника. 1986. № 1. С. 70–76.
17. *Ярмолик В.Н.* // Изв. ВУЗов. Радиоэлектроника. 1986. № 5. С. 53–58.
18. *Ярмолик В.Н.* // АиВТ. 1987. № 5. С. 77–81.

Библиотека БГУИР

19. *Ярмолик В.Н.* // АиВТ. 1987. № 6. С. 42–46.
20. *Yarmolik V.N., Demidenko S.N.* Generation and Application of Pseudorandom Sequences for Random Testing. Chichester, 1988.
21. *Ярмолик В.Н.* Контроль и диагностика цифровых узлов ЭВМ. Минск, 1988.
22. *Ярмолик В.Н.* // Электронное моделирование. 1988. № 6. С. 69–71.
23. *Калоша Е.П., Кацнельсон Е.И., Ярмолик В.Н.* // АиТ. 1989. № 9. С. 160–165.
24. *Ярмолик В.Н., Калоша Е.П.* // Микроэлектроника. 1989. № 3. С. 282–286.
25. *Ярмолик В.Н.* // АиТ. 1989. № 10. С. 159–167.
26. *Yarmolik V.N.* Fault Diagnosis of Digital Circuits. Chichester, 1990.
27. *Ярмолик В.Н., Качан И.В.* // АиВТ. 1990. № 3. С. 89–95.
28. *Ярмолик В.Н., Калоша Е.П.* // АиВТ. 1990. № 1. С. 94–95.
29. *Ярмолик В.Н., Сагалович Ю.Л.* // АиТ. 1990. № 4. С. 155–160.
30. *Калоша Е.П., Качан И.В., Ярмолик В.Н.* // АиТ. 1991. № 1. С. 105–112.
31. *Ярмолик В.Н.* // АиВТ. 1991. № 3. С. 79–83.
32. *Ярмолик В.Н.* // АиТ. 1992. № 1. С. 146–155.
33. *Ярмолик В.Н., Калоша Е.П.* // Электронное моделирование. 1992. № 3. С. 51–56.
34. *Ярмолик В.Н., Быков Ю.В.* // АиВТ. 1992. № 3. С. 63–67.
35. *Yarmolik V.N., Kachan I.V.* Self-Testing VLSI Design. Amsterdam, 1993.
36. *Ярмолик В. Н., Мурашко И.А.* // АиВТ. 1993. № 3. С. 9–13.
37. *Закревский Л.А., Калоша Е.П., Качан И.В., Хаткевич Н.Н., Ярмолик В.Н.* // АиТ. 1994. № 1. С. 3–31.
38. *Ярмолик В.Н., Меметов Г.Р., Николаидис М.* // Микроэлектроника. 1995. № 3. С. 211–215.
39. *Karpovsky M.G., Yarmolik V.N.* // IEEE J. Elec. Test: Theory and Applications. 1996. №.3. P. 251–266.
40. *Ярмолик В.Н., Иванюк А.А.* // Логическое проектирование. 1997. № 2. С. 170–180.
41. *Ярмолик В. Н., Климец Ю. В., Гор Э. Ван де.* // АиВТ. 1997. №5. С. 14–21.
42. *Ярмолик В. Н., Климец Ю. В., Гор Э. Ван де.* // Микроэлектроника. 1997. № 2. С. 266–271.
43. *Мурашко И.А., Шмидман А.М., Ярмолик В.Н.* // АиТ. 1998. № 7. С. 157–167.
44. *Ярмолик В.Н., Иванюк А.А., Янушкевич А.И.* // Микроэлектроника. 1998. Т. 27, № 3. С. 194–199.
45. *Demidenko S, Yarmolik V., Klimets Y. et. al.* // IES J. Elec. and Computer Eng. 1999. № 1. P. 1–5.
46. *Закревский Л.А., Иванюк А.А., Янушкевич А.И., Ярмолик В.Н.* // АиТ. 1999. № 2. С. 120–128.
47. *Иванюк А.А., Янушкевич А.И., Ярмолик В.Н.* // АиТ. 1999. № 1. С. 148–158.
48. *Jarmolik W, Gruszewski M.* // Elektronika. 2001. № 4. P. 26–28.
49. *Ярмолик В.Н., Калоша Е.П., Быков Ю.В. и др.* Проектирование самотестируемых СБИС. Том I. Минск, 2001.
50. *Ярмолик В.Н., Калоша Е.П., Быков Ю.В. и др.* Проектирование самотестируемых СБИС. Том II. Минск, 2001.
51. *Hellebrand S, Wunderlich H.-J., Ivaniuk, A.A. et.al.* // IEEE Trans. on Computer. 2002. № 7. P. 801–809.
52. *Занкович А.П., Ярмолик В.Н.* // АиТ. 2003. № 9. С. 141–154.
53. *Мурашко И.А., Ярмолик В.Н.* Методы минимизации энергопотребления при самотестировании цифровых устройств. Минск, 2004.
54. *Петрик С.Ю., Ярмолик В.Н.* // Информатика. 2004. № 3. С. 58–66.
55. *Мурашко И.А., Ярмолик В.Н.* // АиТ. 2004. № 8. С. 102–114.
56. *Ярмолик В.Н., Мурашко И.А., Куммерт А. и др.* Неразрушающее тестирование запоминающих устройств. Минск, 2005.
57. *Портянко С.С. Ярмолик В.Н.* // Информатика. 2005. № 1. С. 132–138.
58. *Иванюк А.А., Ярмолик В.Н.* Проектирование контролепригодных цифровых устройств. Минск, 2006.
59. *Ярмолик С.В., Ярмолик В.Н.* // Информатика. 2006. № 4. С. 88–96.
60. *Ярмолик В.Н., Портянко С.С., Ярмолик С.В.* Криптография, стеганография и охрана авторского права. Минск, 2007.
61. *Puczko M., Murashko I.A., Yarmolik V.N.* // Miesiecznik naukowo-Techniczny: Pomiary Automatyka Kontrola. 2007. № 7. P. 3–5.
62. *Ярмолик С.В., Ярмолик В.Н.* // АиТ. 2007. № 4. С. 126–137.
63. *Mrozek I, Yarmolik V.N.* // Elektronika. 2007. № 2. P. 28–31.
64. *Иванюк А.А., Ярмолик В.Н.* // АиВТ. 2007. № 3. С. 3–12.
65. *Иванюк А.А., Мусин С.Б., Ярмолик В.Н.* // Микроэлектроника. 2007. Т. 36, № 3. С. 311–318.
66. *Ярмолик С.В., Курбацкий А.Н., Ярмолик В.Н.* // АиВТ. 2008. № 3. С. 15–23.
67. *Иванюк А.А., Ярмолик В.Н.* // АиВТ. 2008. № 4. С. 5–13.
68. *Ярмолик С.В. Курбацкий А.Н. Ярмолик В.Н.* Анализ количественных характеристик различия при тестировании ОЗУ // Информатика. 2008. №3. С. 90–98.
69. *Ivaniuk A.A.* // Radioelectronics and Informatics. 2008. № 4. P. 32–37.
70. *Ярмолик С.В., Занкович А.П., Иванюк А.А.* Маршевые тесты для самотестирования ОЗУ. Минск, 2009.

Библиотека БГУИР

71. *Brzozowski M., Yarmolik V.N.* // Informatyka: Zeszyty Naukowe Politechniki Bialostockiej. 2009. № 4. P. 19–30.
72. *Ярмолик С.В., Ярмолик В.Н.* // АиВТ. 2009. № 4. С. 5–13.
73. *Mrozek I., Yarmolik V.N.* Problemu Funkcjonalnego Testowania Pamieci RAM. Bialystok, 2009.
74. *Ярмолик С.В., Ярмолик В.Н.* // Информатика. 2009. № 3. С. 27–35.
75. *Ярмолик С.В., Ярмолик В.Н.* // АиВТ. 2010. № 4. С. 54–61.
76. *Ярмолик С.В., Ярмолик В.Н.* // Информатика. 2010. № 2. С. 66–75.
77. *Ярмолик С.В., Ярмолик В.Н.* // АиВТ. 2011. № 3. С. 19–30.
78. *Ярмолик С.В., Ярмолик В.Н.* // Информатика. 2011. № 1. С. 79–88.
79. *Ярмолик В.Н., Вашинко Ю.Г.* // Информатика. 2011. № 2. С. 92–103.
80. *Иванюк А.А.* // Информатика. 2011. № 4. С. 113–123.
81. *Ярмолик С.В. Занкович А.П. Иванюк А.А.* Маршевые тесты для тестирования ОЗУ. Гамбург, 2012.
82. *Мурашко И.А., Ярмолик В.Н.* Встроенное самотестирование. Методы минимизации энергопотребления. Гамбург, 2012.
83. *Иванюк А.А.* Проектирование встраиваемых цифровых устройств и систем. Минск, 2012.
84. *Mrozek I., Yarmolik V.N.* // Fundamenta Informaticae. 2012. № 2. P. 1–23.
85. *Mrozek I., Yarmolik V.N.* // IEEE J. Elec. Test: Theory and Applications. 2012. № 3. P. 251–266.
86. *Ярмолик С.В. Ярмолик В.Н.* // АиТ. 2012. №10. С. 142–155.
87. *Ярмолик В.Н., Ярмолик С.В.* // АиВТ. 2013. №5. С. 25–33.
88. *Ярмолик С.В., Ярмолик В.Н.* // Информатика. 2013. № 3. С. 92–103.
89. *Иванюк А.А.* // Информатика. 2013. № 3 (39). С. 82–92.
90. *Заливако С.С., Иванюк А.А.* // Докл. БГУИР. 2013. № 7. С. 37–43.

СВЕДЕНИЯ ОБ АВТОРАХ

Ярмолик Вячеслав Николаевич (1951 г.р.), д.т.н., профессор. В 1973 г.окончил МРТИ. В 1980 г. ему была присвоена ученая степень кандидата технических наук, в 1990 г. – доктора технических наук. Автор 15 монографий и одного учебного пособия, 72 авторских свидетельств на изобретения, более 400 научных работ. Под его научным руководством защищены 3 докторские диссертации и более 20 кандидатских диссертаций. Область научных интересов – тестирование и диагностирование средств вычислительных систем; проектирование самотестируемых встроенных систем и систем на кристалле; синтез контролепригодных вычислительных систем.

Иванюк Александр Александрович, д.т.н., профессор. В 1995 г. окончил БГУИР В 1999 г. ему была присвоена ученуая степень кандидата технических наук, в 2010 г. – доктора технических наук. Автор 7 монографий, 100 научных работ, включая статьи в научных журналах и материалы отечественных и зарубежных научных конференций. Являлся членом организационных комитетов международных научных конференций. В настоящее время – член международного сообщества IEEE Computer Society. Область научных интересов – тестирование и диагностирование средств вычислительных систем.

Библиотека БГУИР