ИНФОРМАТИКА

2017 **январь-март** **№ 1**

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 681.32, 004.056

В.П. Клыбик, С.С. Заливако, А.А. Иванюк

МЕТОД УВЕЛИЧЕНИЯ СТАБИЛЬНОСТИ ФИЗИЧЕСКИ НЕКЛОНИРУЕМОЙ ФУНКЦИИ ТИПА «АРБИТР»

Предлагается метод повышения стабильности физически неклонируемой функции типа «арбитр» без увеличения затрат на аппаратное обеспечение и значительного роста времени получения ответа. Предлагается развернутая параметрическая модель формирования временной разницы тестового сигнала на входах арбитра. Проводится проверка метода на реальных устройствах программируемой логики.

Введение

Повсеместное использование в современных цифровых устройствах программируемых интегральных логических схем (ПЛИС), а также заказных специализированных сверхбольших интегральных схем (СБИС) позволило создать мощные и гибкие решения практически для всех сфер промышленности, быта, медицины, научной и военной отраслей. Увеличение количества таких устройств обусловило актуальность задачи их надежной идентификации. Некоторые способы идентификации цифровых устройств описаны в литературе [1–8]. Одним из перспективных и активно развивающихся способов является использование физически неклонируемых функций (ФНФ) для идентификации цифровых устройств.

По определению, данному в работе [9], физически неклонируемой функцией (от англ. Physical Unclonable Function, *PUF*) является характеристика физической (цифровой) системы, которая не поддается клонированию (копированию, воспроизведению) на других системах. Цифровые системы состоят из компонентов, физические параметры которых на стадии производства принимают случайные, принципиально неуправляемые значения. Наличие случайных параметров делает каждую цифровую систему уникальной и физически неклонируемой. Извлечение уникальных параметров из цифровых систем лежит в основе схемных реализаций ФНФ.

Формально ФНФ описывается значениями пар входных и соответствующих им выходных параметров, которые для цифровых ФНФ являются значениями входных сигналов запроса C(Challenge) и выходных сигналов ответа R (Response). Сама же ФНФ представляет собой функцию преобразования множества запросов C в множество ответов R:

$$R = PUF(C). \quad (1)$$

Существует множество видов ФНФ, детально описанных в статьях [10–16]. Для использования в целях идентификации важным свойством ФНФ является стабильность ответа ФНФ на многократно повторяющийся запрос при одних и тех же условиях. По этим параметрам перспективной для идентификации является ФНФ типа «арбитр» (АФНФ) [10].

Схемная реализация АФНФ состоит из генератора тестового импульса PG и множества N блоков коммутации сигнала (slice), последовательно соединенных в единую цепь и оканчивающихся арбитром (arbiter). В зависимости от управляющего сигнала входной шины запроса $C = \{CH_0, \dots, CH_{N-1}\}$ разрядности N, где CH_i – значение i-го разряда запроса, каждое звено коммутирует два тестовых импульса в прямом или перекрестном направлении (рис. 1).

В зависимости от разности времени прихода двух тестовых импульсов на входы арбитра сигнал на выходе R принимает значение логического нуля или единицы. Разность во времени обусловлена неуправляемыми девиациями производственного процесса при изготовлении интегральной схемы.

Звенья цепи могут быть реализованы на базе мультиплексоров или тристабильных буферных элементов [17]. Второй вариант позволяет достичь меньших аппаратурных затрат, но применим только в заказных СБИС.

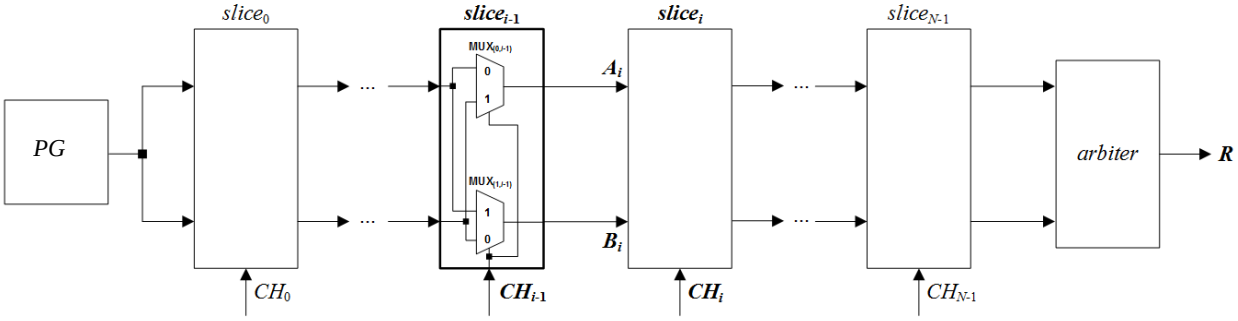

Рис. 1. Структурная схема ФНФ типа «арбитр»

Арбитр может быть реализован на базе как единичных триггеров типа *D* [10], *RS* [18], так и множественных (мультиарбитр) [19]. Одним из вариантов сложного арбитра является составная схема из четырех арбитров [20].

1. Способы стабилизации АФНФ

Варианты реализации арбитра на базе синхронного *D*-триггера или асинхронного *RS*-триггера для стабильной работы требуют наличия определенной разницы во времени прихода фронтов двух тестовых импульсов на входы арбитра. Если же такая разница меньше определенного порогового значения, зависящего от типа и технологии изготовления арбитра, то арбитр оказывается в метастабильном состоянии и значение на его выходе не всегда будет предсказуемым.

Таким образом, возможны ситуации, когда значение на выходе арбитра будет различным для одного и того же запроса при нескольких последовательных экспериментах, другими словами, ответ будет являться нестабильным или ошибочным.

Стабильность ФНФ может быть вычислена с помощью коэффициента ошибок, который определяется как среднее число ошибок (различий), приходящееся на каждый бит ответа ФНФ, полученный в серии экспериментов.

Пусть для проверки стабильности проводится E экспериментов, в результате которых получено n_0 логических нулей ($R = 0$), n_1 логических единиц ($R = 1$), а также n_X метастабильных состояний ($R = X$), под которыми понимается неопределенное значение ответа в терминах значений объектов типа std_logic языка VHDL [21]. Ненулевые значения n_X возможны только в случае реализации арбитра с помощью асинхронного *RS*-триггера.

Обозначим частоту встречаемости ответов 0, 1 и X в E экспериментах как F_0, F_1 и F_X соответственно. Эта величина может быть вычислена следующим образом:

$$F_\alpha = \frac{n_\alpha}{E}, \alpha \in \{0, 1, \mathrm{X}\}. \quad (2)$$

Эталонным ответом R_{ideal} АФНФ на некоторый запрос CH назовем наиболее часто встречающийся в E экспериментах ответ, который может быть вычислен по мажоритарному принципу:

$$R_{ideal} = \begin{cases} 0, \text{если } \max(F_0, F_1, F_X) = F_0; \\ 1, \text{если } \max(F_0, F_1, F_X) = F_1; \\ X, \text{если } \max(F_0, F_1, F_X) = F_X. \end{cases} \quad (3)$$

Обозначим через R_e ответ на запрос CH, полученный в результате эксперимента с индексом $e = 1, 2, \ldots, E$. Тогда стабильность АФНФ $S(CH)$ на запрос CH может быть вычислена как

$$S(CH) = 1 - BER(CH) = 1 - \frac{1}{E}\sum_{e=1}^{E} HD(R_{ideal}, R_e), \quad (4)$$

где *HD* – расстояние Хэмминга, методика расчета которого с учетом метастабильных состояний была представлена в работе [18].

Для ответов, полученных на $K \in [1, 2^N]$ различных запросах CH^{Ω_i} (где Ω_i – десятичное представление *N*-битного двоичного числа $CH = \{CH_{N-1}, \ldots, CH_0\}$, i = 1, 2, …, K), показатель средней стабильности может быть рассчитан с помощью соотношения

$$S_{avg} = \frac{1}{K}\sum_{i=1}^{K} S(CH^{\Omega_i}), \quad (5)$$

а показатель минимальной стабильности – с помощью соотношения

$$S_{min} = \min(S(CH^{\Omega_1}), S(CH^{\Omega_2}), \ldots, S(CH^{\Omega_K})). \quad (6)$$

В работе [18] показано, что для АФНФ с арбитром, реализованным на основе синхронного *D*-триггера и с количеством звеньев N = 128, значения стабильности S_{avg} = 0,5769 и S_{min} = 0,5648.

Существуют специальные подходы для стабилизации ответов АФНФ:

– использование дополнительных корректирующих кодов, получаемых в процессе предварительного обучения системы [22];

– применение детекторов метастабильного состояния арбитра на асинхронном *RS*-триггере [18];

– применение множественного арбитра с комбинаторным анализом выходного алфавита [19];

– мажоритарный анализ ответа для большого числа тестовых импульсов [23].

Все перечисленные подходы требуют либо дополнительных аппаратурных затрат, либо временных издержек на получение стабильного ответа.

В настоящей работе предлагается метод стабилизации АФНФ путем анализа четырех пар «запрос – ответ».

2. Параметрическая модель АФНФ

Для выявления особенностей влияния формирования запросов на ответы АФНФ была разработана детализированная параметрическая модель. Структурный блок *PG* в предлагаемой модели позволяет генерировать тестовый импульс с передним фронтом.

Одно звено схемной реализации АФНФ может быть описано структурным блоком, который имеет следующие входные и выходные порты: A_i, B_i – входные порты для двух копий одного сигнала; C_i, D_i – выходные порты, которые коммутируются с входными портами в зависимости от значения сигнала запроса на входном порту CH_i, $i = 0, \ldots, N-1$. Условимся, что при $CH_i = 0$ происходит коммутация входного порта A_i с выходным портом C_i и входного порта B_i – с выходным портом D_i. В случае когда $CH_i = 1$, происходит перекрестная коммутация входного порта A_i с выходным портом D_i и порта B_i с портом C_i (рис. 2).

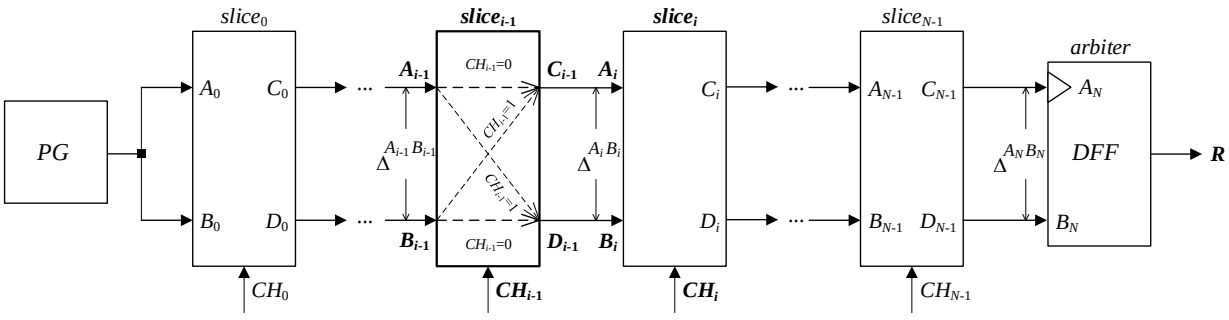

Рис. 2. Обобщенная структура ФНФ типа «арбитр»

Для рассматриваемой модели основными исследуемыми параметрами являются задержки распространения сигналов от входных портов A_i, B_i к выходным портам C_i, D_i в зависимости от значения сигнала на входе CH_i. В связи с этим введем следующее обозначение временной разницы между моментами t^p и t^g наступления фронтов сигналов на портах p и g: $\Delta^{pg} = t^p - t^g$, $p, g \in \{A_i, B_i, C_i, D_i\}$. Например, $\Delta^{A_iB_i} = t^{A_i} - t^{B_i}$ обозначает временную разницу между фронтами сигналов на входных портах A_i и B_i, наступивших в моменты t^{A_i} и t^{B_i} соответственно. Задержка распространения фронта сигнала от входного порта B_i до выходного порта C_i будет иметь обозначение $\Delta^{C_iB_i} = t^{C_i} - t^{B_i}$.

Очевидно, что $\Delta^{pg} = 0, \forall\, p = g$ и $\Delta^{pg} = -\Delta^{gp}, \forall\, p \neq g$.

Согласно определению ФНФ множества значений $\{\Delta^{C_{i-1}A_{i-1}}, \Delta^{D_{i-1}A_{i-1}}, \Delta^{C_{i-1}B_{i-1}}, \Delta^{D_{i-1}B_{i-1}}\}$ и $\{\Delta^{A_iC_{i-1}}, \Delta^{B_iD_{i-1}}\}$, $\forall\, i = 1, \ldots, N$, являются уникальными и неповторяющимися не только для схемной реализации всех звеньев одной схемы на одном полупроводниковом кристалле, но и на множестве кристаллов.

В зависимости от подаваемого значения CH_{i-1} формируются два уникальных маршрута прохождения двух тестовых импульсов от входных портов A_{i-1}, B_{i-1} до портов следующей компоненты A_i, B_i.

При использовании синхронного триггера в качестве схемы арбитра необходимо учитывать его следующие временные параметры: T_S (от англ. setup time) – время предустановки, в течение которого сигнал на входе данных B_N триггера должен оставаться стабильным перед приходом фронта сигнала синхронизации на соответствующий вход A_N; T_H (от англ. hold time) – время удержания, в течение которого сигнал на входе данных триггера должен оставаться стабильным после прихода фронта сигнала синхронизации.

Учет перечисленных параметров для схемы арбитра гарантирует появление стабильного значения ответа на выходе R. В противном случае схема арбитра может оказаться в неопределенном (метастабильном) состоянии, при котором значение ответа на выходе R будет непредсказуемым. В итоге значение ответа на выходе R зависит от результирующей разницы между фронтами сигналов $\Delta^{A_NB_N}$:

$$R = \begin{cases} 0, & \text{если } (-\Delta^{A_NB_N}) \geq T_H; \\ 1, & \text{если } \Delta^{A_NB_N} \geq T_S; \\ X, & \text{если } \Delta^{A_NB_N} < T_S \text{ и } (-\Delta^{A_NB_N}) < T_H. \end{cases} \tag{7}$$

Значение результирующей разницы $\Delta^{A_iB_i}$ формально можно выразить следующей функцией Υ от двух аргументов:

$$\Delta^{A_iB_i} = \Upsilon(\delta_{i-1}^{CH_{i-1}}, \Delta^{A_{i-1}B_{i-1}}), \tag{8}$$

где $\delta_{i-1}^{CH_{i-1}}$ – уникальная характеристика звена $slice_{i-1}$, значение которой зависит от бита запроса CH_{i-1}; $\Delta^{A_{i-1}B_{i-1}}$ – временная разница фронтов сигналов на входе звена $slice_{i-1}$.

Предположим, что $CH_{i-1} = 0$. Тогда значение $\Delta^{A_iB_i}$ можно выразить следующим образом: $\Delta^{A_iB_i} = (t^{A_i} - t^{A_{i-1}}) - (t^{B_i} - t^{B_{i-1}}) + (t^{A_{i-1}} - t^{B_{i-1}}) = \Delta^{A_iA_{i-1}} - \Delta^{B_iB_{i-1}} + \Delta^{A_{i-1}B_{i-1}}$.

Для $CH_{i-1} = 1$ предыдущее выражение принимает вид $\Delta^{A_iB_i} = (t^{A_i} - t^{B_{i-1}}) - (t^{B_i} - t^{A_{i-1}}) + (t^{B_{i-1}} - t^{A_{i-1}}) = \Delta^{A_iB_{i-1}} - \Delta^{B_iA_{i-1}} - \Delta^{A_{i-1}B_{i-1}}$.

Обозначим $\delta_{i-1}^0 = \Delta^{A_iA_{i-1}} - \Delta^{B_iB_{i-1}}$ и $\delta_{i-1}^1 = \Delta^{A_iB_{i-1}} - \Delta^{B_iA_{i-1}}$. Учитывая, что задержки распространения сигналов внутри каждого звена крайне малы и соизмеримы, можно условиться, что $\delta_{i-1}^0 = -\delta_{i-1}^1 = \delta_{i-1}$, при этом $\delta_{i-1}^1 = -\delta_{i-1}^0 = -\delta_{i-1}$, где δ_{i-1} – уникальная характеристика звена $slice_{i-1}$.

Тогда выражение (8) представим в ином виде:

$$\begin{aligned} \Delta^{A_iB_i} &= (1 - CH_{i-1}) \cdot (\delta_{i-1} + \Delta^{A_{i-1}B_{i-1}}) + CH_{i-1} \cdot (-\delta_{i-1} - \Delta^{A_{i-1}B_{i-1}}) = \\ &= (1 - 2CH_{i-1}) \cdot (\delta_{i-1} + \Delta^{A_{i-1}B_{i-1}}). \end{aligned} \tag{9}$$

Таким образом, двоичное значение CH_{i-1} определяет выбор знака для результирующей суммы $(\delta_{i-1} + \Delta^{A_{i-1}B_{i-1}})$.

Для упрощения дальнейших рассуждений введем дополнительную функцию арифметического знака:

$$Sign_{i-1} = 1 - 2CH_{i-1}. \tag{10}$$

Тогда будет справедливо следующее: при $CH_{i-1} = 0$ $Sign_{i-1} = 1$, при $CH_{i-1} = 1$ $Sign_{i-1} = -1$. При этом выражение (9), зависящее от значения CH_{i-1}, принимает вид

$$\Delta^{A_iB_i}(CH_{i-1}) = Sign_{i-1} \cdot \delta_{i-1} + Sign_{i-1} \cdot \Delta^{A_{i-1}B_{i-1}}. \tag{11}$$

Предположим, что в силу отсутствия каких-либо элементов после узла монтажного соединения (см. рис. 2), на котором происходит разделение исходного тестового импульса на два, параметр $\Delta^{A_0B_0} = 0$. Тогда выражение (11) будет определять значение $\Delta^{A_1B_1}$ исключительно исходя из уникальных для $slice_0$ параметров и значения бита запроса CH_0:

$$\Delta^{A_1B_1}(CH_0) = Sign_0 \cdot \delta_0. \tag{12}$$

Выразим значение $\Delta^{A_2B_2}$ исходя из (11) и (12):

$$\Delta^{A_2B_2}(CH_1, CH_0) = Sign_1 \cdot \delta_1 + Sign_1 \cdot \Delta^{A_1B_1} = Sign_1 \cdot \delta_1 + Sign_1 \cdot Sign_0 \cdot \delta_0. \tag{13}$$

Из (13) видно, что $\Delta^{A_2B_2}$ зависит не только от значения CH_1, но и от значения предыдущего разряда запроса CH_0. В связи с этим левую часть равенства (13) можно представить как $\Delta^{A_2B_2}(CH_1, CH_0)$.

Вычислим значение

$$\Delta^{A_3B_3}(CH_2, CH_1, CH_0) = Sign_2 \cdot \delta_2 + Sign_2 \cdot Sign_1 \cdot \delta_1 + Sign_2 \cdot Sign_1 \cdot Sign_0 \cdot \delta_0. \tag{14}$$

Обобщая, можно записать выражение

$$\begin{gathered}\Delta^{A_NB_N}(CH_{N-1}, \dots, CH_0) = \Delta^{A_NB_N}(CH^{\Omega}) = \\ = Sign_{N-1} \cdot \delta_{N-1} + Sign_{N-1} \cdot Sign_{N-2} \cdot \delta_{N-2} + \cdots + Sign_{N-1} \cdot \dots \cdot Sign_0 \cdot \delta_0.\end{gathered} \tag{15}$$

Например, для $CH^{\Omega} = \{CH_{N-1}, CH_{N-2}, CH_{N-3,} \dots, CH_0\} = \{0,\ 0,\ 0, \dots,\ 0\} = CH^0$ предыдущее выражение (15) можно представить как

$$\Delta^{A_NB_N}(CH^0) = \delta_{N-1} + \delta_{N-2} + \cdots + \delta_0 = \sum_{i=0}^{N-1} \delta_i. \tag{16}$$

Рассмотрим еще несколько примеров для различных значений запросов. Предположим, что запрос принимает значение CH^1, т. е. $CH_0 = 1$, а $CH_{\text{i}} = 0, \forall\, i = 1, \dots, N-1$:

$$\Delta^{A_NB_N}(CH^1) = \delta_{N-1} + \delta_{N-2} + \cdots - \delta_0 = \sum_{i=0}^{N-1} \delta_i - 2 \cdot \delta_0 = \Delta^{A_NB_N}(CH^0) - 2 \cdot \delta_0. \tag{17}$$

Допустим, что для CH^0 схема АФНФ вырабатывает стабильное значение ответа на выходе R (см. выражение (7)). С учетом того что значение $2 \cdot \delta_0$ крайне мало в сравнении со значением $\Delta^{A_NB_N}(CH^0)$, вероятность изменения значения ответа для CH^1 также крайне мала.

Пусть $CH^{2^{N-1}} = \{1,\ 0,\ 0, \dots,\ 0\}$, тогда

$$\Delta^{A_NB_N}\left(CH^{2^{N-1}}\right) = -\delta_{N-1} - \delta_{N-2} - \cdots - \delta_0 = -\sum_{i=0}^{N-1} \delta_i = -\Delta^{A_NB_N}(CH^0), \tag{18}$$

что означает высокую вероятность инверсии стабильного ответа R, полученного при запросе CH^0.

Рассмотрим запросы $CH^{\Omega}, CH^{\Omega'}, CH'^{\Omega}, CH'^{\Omega'}$, для которых выполняются равенства $|\Omega - \Omega'| = 1,\ |\Omega - '\Omega| = 2^{N-1},\ |\Omega - '\Omega| = 2^{N-1},\ |'\Omega' - '\Omega| = 1,\ |'\Omega' - \Omega'| = 2^{N-1}$. Выполнение указанных равенств означает, что относительно запроса CH^{Ω} для запроса $CH^{\Omega'}$ будет инвертирован младший бит запроса $LSB = CH_0$, для запроса CH'^{Ω} – старший бит запроса $MSB = CHN-1$, а для запроса $CH'\Omega'$ будут инвертированы и младший, и старший биты *LSB* и *MSB*.

Рассмотрим изменения значения $\Delta^{A_N B_N}(CH^{\Omega})$ при инверсии *LSB* и *MSB* согласно формуле (15).

При инверсии *LSB* изменяется значение запроса CH^{Ω} на $CH^{\Omega'}$ ($|\Omega - \Omega'| = 1$) и происходит изменение знака компоненты δ_0 на противоположный согласно (17).

Оценим следующую разницу:

$$\begin{gathered}\Delta(\overline{LSB}) = |\Delta^{A_N B_N}\left(CH^{\Omega}\right) - \Delta^{A_N B_N}\left(CH^{\Omega'}\right)| = \\ = |Sign_{N-1} \cdot \ldots \cdot Sign_1 \cdot \delta_0 \cdot \left(Sign_0 - (-Sign_0)\right)| = |2 \cdot \delta_0|\,.\end{gathered} \tag{19}$$

Если для произвольного запроса CH^{Ω} верно неравенство $\Omega < 2^{N-1}$, то $MSB = 0$ и $Sign_{N-1} = 1$ согласно (15), тогда

$$\Delta^{A_N B_N}\left(CH^{\Omega}\right) = \delta_{N-1} + Sign_{N-2} \cdot \delta_{N-2} + \cdots + Sign_{N-2} \cdot \ldots \cdot Sign_0 \cdot \delta_0. \tag{20}$$

Для запроса CH'^{Ω}, где $'\Omega = \Omega + 2^{N-1}$, будут верны равенства $MSB = 1$ и $Sign_{N-1} = -1$, при этом

$$\begin{gathered}\Delta^{A_N B_N}\left(CH'^{\Omega}\right) = -(\delta_{N-1} + Sign_{N-2} \cdot \delta_{N-2} + \cdots + \\ + Sign_{N-2} \cdot \ldots \cdot Sign_0 \cdot \delta_0).\end{gathered} \tag{21}$$

Таким образом,

$$\Delta^{A_N B_N}\left(CH^{\Omega}\right) = -\Delta^{A_N B_N}\left(CH'^{\Omega}\right). \tag{22}$$

Оценим модуль разности

$$|\Delta^{A_N B_N}\left(CH^{\Omega}\right) - \Delta^{A_N B_N}\left(CH'^{\Omega}\right)| = 2 \cdot |\Delta^{A_N B_N}\left(CH^{\Omega}\right)|\,. \tag{23}$$

Из формул (19)–(23) видно, что инверсия *LSB* для произвольного запроса способна внести относительно малые изменения в общее значение задержки сигналов, а инверсия *MSB* приводит к инверсии знака результата. При этом для произвольной длины путей присутствует отличная от нуля вероятность существования запросов, которые сформируют результирующую задержку $\Delta^{A_N B_N} \approx 0$ ввиду различных значений и знаков задержек на каждом звене. Последнее означает, что всегда существует подмножество запросов из множества всех возможных, генерирующее нестабильные ответы арбитра $R = X$, тогда как для получения стабильных ответов арбитра должны выполняться следующие условия:

$$R = \begin{cases} 0, \text{если } -\Delta^{A_N B_N} \gg T_H; \\ 1, \text{если } \Delta^{A_N B_N} \gg T_S. \end{cases} \tag{24}$$

Для идеального арбитра верно выполнение равенства $|T_H| = |T_S|, T_S \to 0$.

Как следствие инверсия *LSB* для произвольного запроса имеет малую вероятность инверсии ответа идеального арбитра, а инверсия *MSB* – высокую вероятность инверсии ответа идеального арбитра.

Если инверсия *MSB* произвольного запроса не привела к инверсии ответа, то можно утверждать, что $\Delta^{A_N B_N}$ для этого запроса находится в зоне метастабильности арбитра. Для реального арбитра отмечается сдвиг зоны метастабильности относительно нуля по причине наличия асимметрии при технологической реализации арбитра и выполнение неравенства $|T_H| \neq \neq |T_S|,\ T_H \neq 0, T_S \neq 0$. Например, для параметрической модели арбитра, реализованной на

FPGA Xilinx Artix-7, параметры предустановки и удержания принимают значения $T_H = -0{,}103$ нс, $T_S = 0{,}297$ нс.

Рассмотрим последовательность ответов $\{R_0, R_1, R_2, R_3\}$, где $R_0 = PUF(CH^{\Omega})$, $R_1 = PUF(CH^{\Omega'})$, $R_2 = PUF(CH'^{\Omega})$, $R_3 = PUF(CH'^{\Omega'})$.

Введем понятие сильных запросов, для которых верно $R_0 = R_1 = \overline{R_2} = \overline{R_3}$. В результате для них могут быть получены только два варианта последовательности ответов: {1,1,0,0} либо {0,0,1,1}. Все остальные запросы, сформированные по описанному выше принципу и дающие иную последовательность ответов, будем называть слабыми.

Анализируя последовательность из четырех ответов, полученных в результате подачи четырех запросов, которые сформированы описанным способом, можно выявить сильные и слабые запросы и для них с высокой вероятностью прогнозировать стабильность ответа АФНФ на любой из поданных запросов. Только две последовательности ответов {1,1,0,0} и {0,0,1,1} являются сигнатурами сильных запросов и предполагают стабильность ответа, поэтому для формирования идентификаторов цифрового устройства необходимо использовать только ответы на сильные запросы.

3. Исследование параметрической модели АФНФ

Для проверки разработанной математической модели было создано VHDL-описание схемной реализации АФНФ из $N = 10$ звеньев. Проект VHDL был реализован в виде параметрической post place&route-модели для кристалла FPGA Xilinx Artix-7. Также было создано тестовое окружение, позволяющее автоматизировать процессы формирования последовательности запросов, анализа ответа арбитра R и числового значения задержки $\Delta^{A_N B_N}$. В ходе поставленного эксперимента было сгенерировано множество всех $2^N = 1024$ запросов.

Рассмотрим график отсортированных по возрастанию значений $\Delta^{A_N B_N}$ R и значения ответа R, включая неопределенные значения X, соответствующие метастабильному состоянию арбитра, в зависимости от подаваемых запросов (рис. 3).

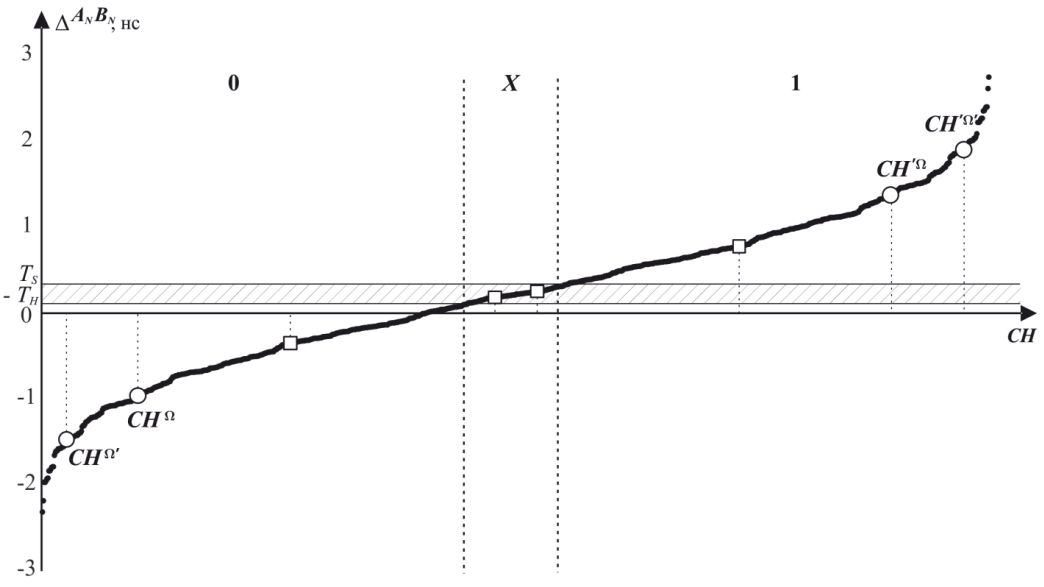

Рис. 3. Расположение групп сильных и слабых запросов

Как видно из представленного графика, зависимость $\Delta^{A_N B_N}$ от значения подаваемого запроса является нелинейной, что обусловлено наличием на звеньях задержек δ_i с разными значениями и разными знаками. Помимо этого из графика видно, что зона метастабильности реализованного арбитра смещена в сторону положительных значений относительно идеального арбитра.

На рис. 3 показана также группа слабых запросов, обозначенных символом □, и группа сильных запросов, обозначенных символом ○. Группа слабых запросов {26, 27, 538, 539} формирует на арбитре задержки {0,182, –0,346, 0,252, 0,78} нс и дает множество ответов арбитра {X, 0, X, 1}. Группа сильных запросов {56, 57, 568, 569} формирует на арбитре задержки {–0,945, –1,473, 1,379, 1,907} нс и дает множество ответов арбитра {0, 0, 1, 1}.

Разделим ось абсцисс на три региона в зависимости от ответов арбитра на соответствующий запрос. Видно, что слабые запросы расположены на графике ближе друг к другу, чем сильные, а также лежат вблизи региона метастабильности арбитра или внутри него.

Для обозначения границ метастабильного региона были выбраны значения $-T_H$ и T_S, соответствующие параметрам арбитра. Действительно, в соответствии с формулой (7) арбитр попадает в метастабильное состояние при условии $\Delta^{A_N B_N} < T_S$ и $(-\Delta^{A_N B_N}) < T_H$, которое эквивалентно выражению $-T_H < \Delta^{A_N B_N} < T_S$.

Анализ экспериментальных данных, полученных на параметрической модели АФНФ, показал преобладание сильных групп запросов: 67 % сильных и 33 % слабых.

4. Результаты экспериментов

4.1. Реализация экспериментальной установки

Для проведения натурных экспериментов была выбрана модель ПЛИС, отличная от той, на которой проводилось параметрическое моделирование. Такое решение позволяет показать, что разработанная математическая модель АФНФ может быть реализована на произвольной ПЛИС, поэтому в качестве платформы для реализации была выбрана система на кристалле ZC706 (СнК), входящая в состав платы быстрого прототипирования Xilinx Zynq-7000 [24].

Для экспериментальной проверки предлагаемого метода была реализована АФНФ с N = 130 звеньями и арбитром на базе синхронного *D*-триггера, а также асинхронного *RS*-триггера (рис. 4).

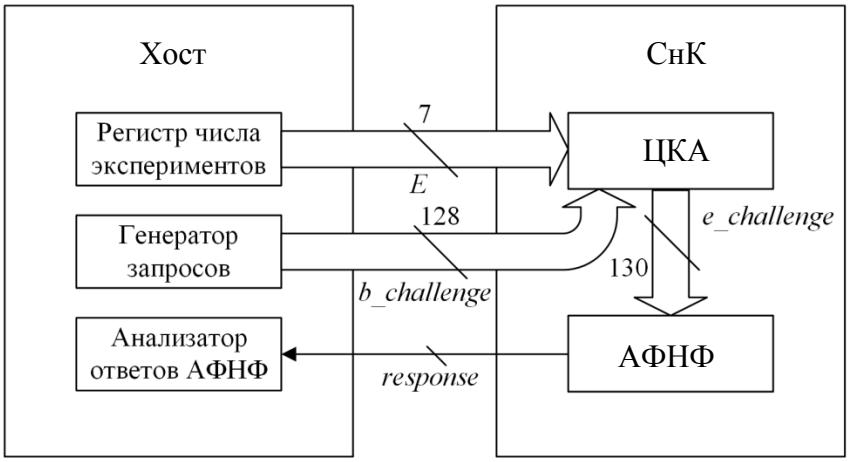

Рис. 4. Структурная схема реализованного устройства

С помощью пакета Xilinx SDK было разработано программное обеспечение на языке C для взаимодействия с АФНФ, реализованной в среде Xilinx Vivado 2014.4.1 на языке VHDL. На стороне хоста был реализован генератор псевдослучайной М-последовательности для получения 128-битных слабо коррелированных запросов, а также возможность с помощью регистра задать количество экспериментов E (от 1 до 128) по получению ответа ФНФ на фиксированный запрос. Базовый запрос (*b_challenge*) подавался на вход цифрового конечного автомата (ЦКА), который, в свою очередь, осуществлял запуск E экспериментов по получению ответов вида $\{R_0, R_1, R_2, R_3\}$.

Таким образом, экспериментальная установка позволила произвести тестирование фиксированной аппаратной конфигурации АФНФ на произвольном числе запросов.

4.2. Распределение конфигураций ответов

Для тестирования АФНФ с арбитром на базе *D*-триггера был проведен эксперимент по генерированию $K = 10\,000$ запросов, повторенных $E = 100$ раз, по описанному выше принципу. Распределение конфигураций ответов на сильные и слабые запросы показано на рис. 5.

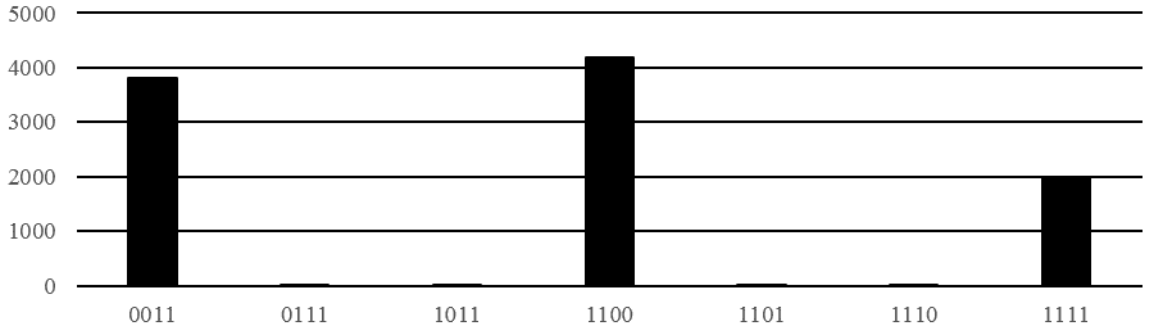

Рис. 5. Распределение конфигураций ответов на сильные и слабые запросы для АФНФ с арбитром на базе *D*-триггера

Из диаграммы видно, что ответы на сильные запросы ({0, 0, 1, 1} и {1, 1, 0, 0}) составляют порядка 80 % от всех сгенерированных. Наличие ответов вида {1, 1, 1, 1} свидетельствует о том, что регион метастабильности занимает часть региона логической единицы. Следовательно, наблюдается преобладание значений ответов {1, 1, 0, 0} над значениями {0, 0, 1, 1}.

Для проверки стабильности генерируемых ответов был реализован арбитр на базе асинхронного *RS*-триггера, с помощью которого было также сгенерировано $K = 10\,000$ запросов, повторенных $E = 100$ раз. Распределение конфигураций ответов на эти запросы показано на рис. 6.

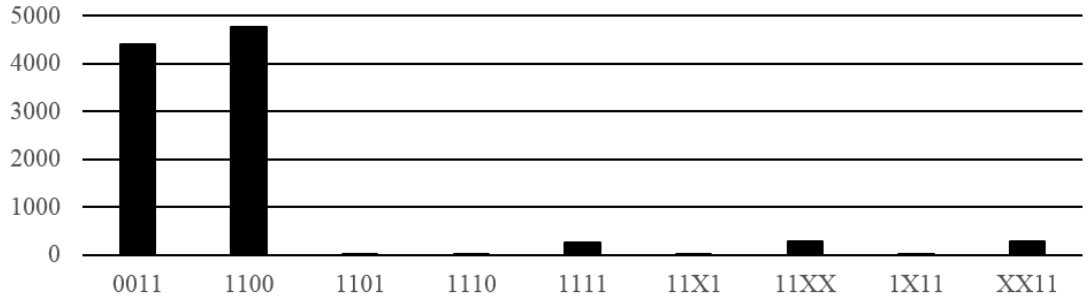

Рис. 6. Распределение конфигураций ответов на сильные и слабые запросы для АФНФ с арбитром на базе *RS*-защелки

Подобная реализация АФНФ показала несколько другое распределение запросов: 90 % сильных и 10 % слабых в силу способности арбитра, реализованного на базе асинхронного *RS*-триггера, обнаруживать метастабильное состояние. Также множество возможных ответов имеет бóльшую мощность, поскольку в него входят другие конфигурации, один или несколько ответов которых содержат метастабильное состояние *X*.

Различия в распределении сильных и слабых запросов между реализацией на ПЛИС и параметрическим моделированием объясняются принципиальной невозможностью моделирования реальных значений временных задержек и параметров арбитра. Тем не менее результаты экспериментов подтверждают тенденцию преобладания сильных запросов над слабыми.

Показатели стабильности для всех возможных ответов (S_{avg} и S_{min}), а также оценка вероятности получения стабильного ответа (P_{stable}) приведены в таблице.

Стабильность конфигураций ответов АФНФ с арбитром на базе асинхронного *RS*-триггера

$\{R_0, R_1, R_2, R_3\}$	S_{avg}	S_{min}	P_{stable}
{1, 1, 0, 0}	**0,999**	**0,758**	**0,991**
{0, 0, 1, 1}	**0,999**	**0,768**	**0,993**
$\{1, 1, X, X\}$	0,972	0,753	0,651
$\{X, X, 1, 1\}$	0,976	0,775	0,669
$\{1, 1, 1, 1\}$	0,944	0,758	0,279
$\{0, 1, 1, 1\}$	0,778	0,778	0,000
$\{1, 1, 0, 1\}$	0,740	0,733	0,000
$\{1, 1, X, 1\}$	0,774	0,763	0,000
$\{1, 1, 1, 0\}$	0,753	0,753	0,000
$\{1, 1, 1, X\}$	0,776	0,735	0,000
$\{X, 1, 1, 1\}$	0,766	0,755	0,000
$\{1, X, 1, 1\}$	0,741	0,718	0,000

Примечание: выделены конфигурации ответов, имеющие хорошую среднюю стабильность.

Понятие вероятности появления стабильного ответа P_{stable} было введено для сравнения качества ответов АФНФ, полученных с помощью предлагаемого метода и без него. Численное значение этого показателя может быть оценено с помощью выражения

$$P_{stable} = \frac{K_{stable}}{K}, \tag{25}$$

где K_{stable} – количество ответов, стабильность которых $S(CH) = 1$.

В таблице показано, что конфигурации ответов {1, 1, 0, 0} и {0, 0, 1, 1} имеют хорошую среднюю стабильность, однако минимальная находится на таком же уровне, как и при слабых запросах. Этот эффект объясняется тем, что если ответ АФНФ является нестабильным, то показатель его стабильности примерно одинаков (≈0,75) как для слабых, так и для сильных запросов, однако вероятность встретить данный ответ для сильных запросов незначительна (менее 0,01).

Таким образом, поскольку количество сильных запросов достаточно велико (9161 из 10 000), то оценка вероятности может быть признана состоятельной и такое же соотношение стабильных ответов к нестабильным будет наблюдаться и при бóльшем числе запросов.

4.3. Генерирование идентификатора

Для генерирования идентификатора был разработан следующий алгоритм:

1. Тестирование запроса.

1.1. Выдвигается гипотеза о сильном запросе: однократно подаются четыре запроса вида CH^{Ω}, CH'^{Ω}, $CH^{\Omega'}$, $CH'^{\Omega'}$ к ФНФ и считываются ответы на них $\{R_0, R_1, R_2, R_3\}$.

1.2. Из всех сгенерированных конфигураций ответов выбираются только те, которые имеют вид {0, 0, 1, 1} или {1, 1, 0, 0}, и соответствующие им запросы помечаются как сильные.

1.3. Остальные запросы помечаются как слабые.

2. Генерирование бита идентификатора.

2.1. Из сильных запросов выбирается такие, которые ранее не были использованы для генерирования идентификатора (в последовательности, определенной генератором запросов).

2.2. Ответ генерируется E раз, в результате чего проверяется его стабильность.

Был проведен эксперимент по генерированию 25 (20 на основе сильных и 5 на основе слабых запросов) 128-битных идентификаторов, отобранных из 4000 запросов. В результате средняя стабильность идентификаторов, построенных на сильных запросах, лежит в интервале [0,996, 1,000], а идентификаторов на основе слабых запросов – в интервале [0,743, 0,779]. Следовательно, предлагаемый способ генерирования идентификаторов обладает гораздо более высокой средней стабильностью (0,999), чем генерирование без учета свойств запроса (0,759).

Также был проведен эксперимент по генерированию 200 идентификаторов разной длины L (L = 1, ..., 128 бит). Результаты эксперимента показаны на рис. 7 (график выполнен в логарифмическом масштабе по оси ординат и в линейном масштабе по оси абсцисс).

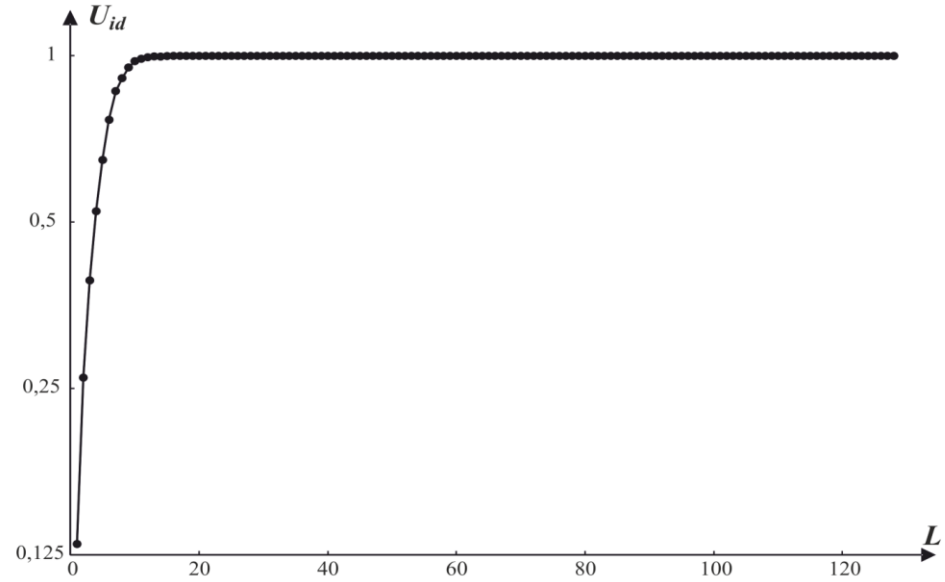

Рис. 7. Значение уникальности идентификатора U_{id} в зависимости от его длины L

Определим уникальность идентификатора как

$$U_{id} = 1 - \frac{Id_{avg}}{K_{id}}, \tag{26}$$

где Id_{avg} – среднее число идентификаторов, приходящееся на его определенное значение (например, для идентификаторов длиной 1 количество идентификаторов со значением 0 – 118, а со значением 1 – 82, следовательно, среднее значение 100); K_{id} – количество идентификаторов, которое было сгенерировано. В данном случае K_{id} = 200.

По результатам эксперимента удалось установить, что для идентификаторов длиной 16 и более все K_{id} = 200 значений являются уникальными (т. е. не повторяются), а для идентификаторов меньшей разрядности значение уникальности экспоненциально возрастает с увеличением длины. При этом расстояние Хэмминга между идентификаторами, нормированное по их длине, принимает значения в интервале [0,479, 0,495] и не зависит от разрядности идентификатора.

Таким образом, предлагаемый способ позволяет генерировать уникальные идентификаторы разрядностью 16 и более со значительно повышенной стабильностью без дополнительных аппаратурных затрат за приемлемое время.

Заключение

Разработанная математическая модель показывает, что для классической АФНФ всегда существует подмножество запросов, ответы на которые будут нестабильны. Стабильность ответа АФНФ на любой запрос можно с высокой вероятностью предсказать путем анализа ответов на три дополнительных запроса, сформированных в результате изменения *LSB* и *MSB*. Соответственно эффективное количество пар «запрос – ответ» для АФНФ из N звеньев не превышает 2^{N-2}.

Результаты параметрического моделирования на кристалле FPGA Xilinx Artix-7 показали, что множество запросов к ФНФ может быть разделено на 67 % сильных и 33 % слабых, что подтверждает выдвинутую гипотезу. Предлагаемая модель была протестирована с помощью экспериментальной установки, реализованной на плате быстрого прототипирования Xilinx Zynq-7000. Результаты экспериментов с реализацией арбитра при помощи синхронного

D-триггера показали соотношение слабых и сильных запросов 80 на 20 %, а с реализацией с помощью асинхронного *RS*-триггера – 90 на 10 % соответственно. Экспериментальные данные подтвердили гипотезу о преобладании сильных запросов над слабыми. Значения средней стабильности, а также вероятности появления стабильных ответов для сильных запросов значительно (более чем на 25 %) превышают такие же показатели, полученные для слабых запросов.

Был разработан алгоритм генерирования уникального идентификатора с помощью АФНФ, основанный на применении предложенной математической модели. Алгоритм позволяет генерировать уникальные идентификаторы разрядностью более 16 бит с повышенной стабильностью и незначительными временными издержками.

Список литературы

1. Pentium III Serial Numbers [Electronic resource]. – Mechler Enterprises, LLC., 2001. – Mode of access : http://www.pcmech.com/article/pentium-iii-serial-numbers. – Date of access : 30.05.2016.

2. Silicon physical random functions / B. Gassend [et al.] // ACM Conf. on Comp. and Comm. Security (CCS'02), N. Y., USA, November 18–22, 2002 / ACM N. Y. – N. Y., 2002. – P. 148–160.

3. Koushanfar, F. Intellectual property metering / F. Koushanfar, G. Qu, M. Potkonjak // 4th Intern. Workshop Information Hiding (IH'01), Pittsburg, USA, April 25–27, 2001. – Pittsburg, 2001. – P. 81–95.

4. Koushanfar, F. Hardware Metering / F. Koushanfar, G. Qu // Proc. IEEE Design Automation Conf. (DAC'01), Scottsdale, USA, June 18–22, 2001 / ACM N. Y. – Scottsdale, 2001. – P. 490–493.

5. Qu, G. Fingerprinting intellectual property using constraint-addition / G. Qu, M. Potkonjak // Proc. IEEE Design Automation Conf. (DAC'00), Los-Angeles, USA, June 5–9, 2000 / ACM, N. Y. – Los-Angeles, 2000. – P. 587–592.

6. A blind dynamic fingerprinting technique for sequential circuit intellectual property protection / C.-H. Chang [et al.] // IEEE Trans. Comput.-Aided Design of Integrated Circuits and Syst. – 2014. – Vol. 33, no. 1. – P. 76–89.

7. Koushanfar, F. Provably secure active IC metering techniques for piracy avoidance and digital rights management / F. Koushanfar // IEEE Trans. Inf. Forensics and Security. – 2011. – Vol. 7, no. 1. – P. 51–63.

8. Ending piracy of integrated circuits / J. Roy [et al.] // Computer. – 2010. – Vol. 43, no. 10. – P. 30–38.

9. Architecture and Design Flow for a Highly Efficient Structured ASIC / H. Man-Ho [et al.] // IEEE Transactions on VLSI Systems. – 2012. – Vol. 21, iss. 3. – P. 424 – 433.

10. A technique to build a secret key in integrated circuits for identification and authentication applications / J.W. Lee [et al.] // Intern. Symp. VLSI Circuits (VLSI'04), Honolulu, USA, June 15–19, 2004. – Honolulu, 2004. – P. 176–179.

11. Readproof hardware from protective coatings / P. Tuyls [et al.] // Cryptographic Hardware and Embedded Systems (CHES'06), Yokohama, Japan, October 10–13, 2006 / Springer, N. Y. – Yokohama, 2006. – P. 369–383.

12. Holcomb, D.E. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags / D.E. Holcomb, W.P. Burleson, K. Fu // Conf. RFID Security, Malaga, Spain, July 11–13, 2007 / University of Malaga, Spain. – Malaga, 2007. – P. 1–2.

13. The butterfly PUF protecting IP on every FPGA / S.S. Kumar [et al.] // Proc. Intern. Workshop Hardware-Oriented Security and Trust (HOST'08), Anaheim, USA, June 8–10, 2008. – Anaheim, 2008. – P. 67–70.

14. Uniqueness enhancement of PUF responses based on the locations of random outputting RS latches / D. Yamamoto [et al.] // Cryptographic Hardware and Embedded Systems (CHES'11), Nara, Japan, September 28 – October 1, 2011. – Nara, 2011. – P. 390–406.

15. Ярмолик, В.Н. Физически неклонируемые функции / В.Н. Ярмолик, Ю.Г. Вашинко // Информатика. – 2011. – № 2. – С. 92–103.

16. CMOS image sensor based physical unclonable function for coherent sensor-level authentication / Y. Cao [et al.] // IEEE Trans. Circuits and Systems I: Regular Papers. – 2015. – Vol. 62, no. 11. – P. 2629–2640.

17. Ozturk, E. Physical unclonable function with tristate buffers / E. Ozturk, G. Hammouri, B. Sunar // Proc. Intern. Symp. on Circ. and Syst. (ISCAS'08), Seattle, USA, May 18–21, 2008. – Seattle, 2008. – P. 3194–3197.

18. Multi-valued arbiters for quality enhancement of PUF responses on FPGA implementation / S.S. Zalivaka [et al.] // Special Session on Cyber-Physical Systems and Security, in Proc. 21st IEEE Asia and South Pacific Design Automation Conf. (ASP-DAC 2016), Macao, China, 26–28 Jan., 2016. – Macao, 2016. – P. 533–538.

19. Клыбик, В.П. Применение физически неклонируемой функции типа арбитр для решения задачи идентификации цифровых устройств / В.П. Клыбик, А.А. Иванюк // Автоматика и вычислительная техника. – 2015. – № 3(49). – С. 24–34.

20. Иванюк, А.А. Использование физически неклонируемых функций типа «арбитр» для идентификации встроенных систем на базе ПЛИС / А.А. Иванюк // Информационные технологии и системы 2011 (ИТС 2011) : материалы Междунар. науч. конф., БГУИР, Минск, Беларусь, 26 окт. 2011 г. / редкол. : Л.Ю. Шилин [и др]. – Минск : БГУИР, 2011. – С. 270–271.

21. 1076-2008 – IEEE Standard VHDL Language Reference Manual [Electronic resource]. – IEEE., Jan. 2009. – Mode of access : http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4772740&filter=AND(p_Publication_Number:4772738). – Date of access : 30.05.2016.

22. Maes, R. PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator / R. Maes, A. van Herrewege, I. Verbauwhede // Cryptographic Hardware and Embedded Systems (CHES'12), Leuven, Belgium, September 9–12, 2012. – Leuven, 2012. – P. 302–319.

23. Иванюк, А.А. Аппаратная реализация алгоритма идентификации ПЛИС на основе физически неклонируемых функций / А.А. Иванюк // Информационные технологии и системы 2013 (ИТС 2013) : материалы Междунар. науч. конф., БГУИР, Минск, Беларусь, 23 окт. 2013 г. / редкол. : Л.Ю. Шилин [и др]. – Минск : БГУИР, 2013. – С. 184–271.

24. Zynq-7000 All Programmable SoC Overview (DS190) – Xilinx [Electronic resource]. – Xilinx, Inc., 2012. – Mode of access : http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf. – Date of access : 30.05.2016.

Поступила 23.06.2016

Белорусский государственный университет информатики и радиоэлектроники,
Минск, ул. П. Бровки, 6
e-mail: klybik@bsuir.by,
zalivako@bsuir.by,
ivaniuk@bsuir.by

V.P. Klybik, S.S. Zalivaka, A.A. Ivaniuk

RELIABILITY ENHANCEMENT METHOD FOR «ARBITER» PHYSICALLY UNCLONABLE FUNCTION

The paper presents a reliability enhancement method for an arbiter physically unclonable function (A-PUF). The proposed technique has reasonable challenge-response generation time and does not cause additional hardware overheads. A time difference of a test pulse delay has been used as a basis for A-PUF parametric model development. The proposed approach has been verified on a real programmable logic device.