

---

УДК 519.1(075.8)+510.6(075:8)

С.Ф. Тюрин<sup>1</sup>, В.С. Харченко<sup>2</sup>

<sup>1</sup> Пермский национальный исследовательский политехнический университет, Россия

<sup>2</sup> Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

## «ЗЕЛЁНАЯ» ПРОГРАММИРУЕМАЯ ЛОГИКА: КОНЦЕПЦИЯ И ЭЛЕМЕНТЫ РЕАЛИЗАЦИИ ДЛЯ FPGA ПРОЕКТОВ

*Анализируются методы энергосбережения для проектов программируемой логики на FPGA кристаллах. Предлагаются концепция «зеленой» программируемой логики для FPGA проектов и элементы ее реализации на неизбыточных и толерантных к отказам базисах. Исследуются варианты создания энергоэффективной логики FPGA на основе программно-аппаратной реализации булевых функций и автоматов и перераспределения функций между блоками таких реализаций с учётом необходимости парирования сбоев при понижении напряжения питания.*

**Ключевые слова:** *зелёная логика, программируемая логическая интегральная схема, FPGA, конфигурируемый логический блок, логический элемент, Look Up Table – LUT, сбоеустойчивость.*

### Введение

**Мотивация:** «зеленые» вычисления. В настоящее время на «зелёной» волне борьбы за защиту и сохранение окружающей среды, экономию ресурсов, прежде всего, энергетических, в мире развернулись и активно ведутся научные исследования и разрабатываются технологии в области так называемого «green computing» и «green IT» («зеленого компьютеринга» и «зеленых ИТ»), в которых компьютерные и коммуникационные средства рассматриваются как объект снижения энерго- и ресурсопот-

ребления в целом [1, 2]. Актуальность вопроса состоит в том, что эти средства, являясь инструментом управления энергосберегающими системами, сами потребляют до 3% вырабатываемой энергии. Разработана концепция «Energy modulated computing» [3]. Монография [2], вышедшая в 2013 г., стала первой работой, обобщившей некоторый опыт в данной области. Растет число конференций и семинаров, посвященных данной тематике [4].

В ноябре 2012 года стартовал 3-летний международный проект TEMPUS-GreenCo «Green Computing and Communications» [5, 6], выполняемый уни-

верситетами и НЦ Украины, Великобритании, Греции, Италии, Словакии и России, координируемый кафедрой компьютерных систем и сетей Национального аэрокосмического университета «ХАИ».

Задачей проекта является разработка уникальной образовательного комплекса, включая 14 магистерских, докторантских и тренинг-курсов, объединяющих научно-технологические компоненты по ресурсосбережению в кристаллах микропроцессоров, программируемых логических интегральных схем (ПЛИС), сетевого оборудования, беспроводных и мобильных систем, дата-центров, разработке «зеленого» программного обеспечения, web- и cloud-систем, менеджменту создания и реинжиниринга ИТ-инфраструктур [6]. Одна из составляющих проекта связана с разработкой курсов по энергоэффективному, «зеленому» ПЛИС типа FPGA (Green FPGA). Энергоэффективность будет пониматься в смысле работы [3] как объем вычислений на единицу использованной энергии.

**Анализ методов энергосбережения для FPGA проектов.** Методы энергосбережения при проектировании и реализации систем на FPGA, описаны в [2, 3, 7 – 9]. Они базируются на таких принципах:

- режимной адаптации, при которой производится управляемое отключение (переход в «спящий» режим) всего кристалла или его частей;
- работе на пониженном напряжении при сохранении допустимого уровня сбоев вследствие нештатном питании;
- уменьшения сложности проекта и его автоматной декомпозиции по критерию энергозатрат;
- диверсной синхронизации различных частей проекта (каналов), позволяющей уменьшить число одновременно срабатывающих элементов;
- минимизации коммуникационных ресурсов проекта на уровне логических ячеек и др.

Следует подчеркнуть, что FPGA – это наиболее удобная технология для демонстрации и внедрения принципов «зеленого» дизайна на вентиляном и логическом уровнях. В то же время эти уровни недостаточно исследованы для существующих и перспективных семейств FPGA. В свете этих трендов представляется целесообразным исследовать возможности создания своего рода «зеленой», энергосберегающей логики, ориентированной на FPGA.

**Цель статьи** – разработка концепции и элементов реализации «зеленой» программируемой логики на FPGA на избыточных и толерантных к отказам базисах. Статья имеет следующую структуру: в первом разделе анализируется существующая логика FPGA в контексте «зеленых» вычислений, во втором – формулируются основные положения концепции «зеленой» программируемой логики; третий раздел посвящен особенностям программно-аппаратной реализации проектов FPGA, в т.ч. и с ис-

пользованием функционально полных толерантных (ФПТ) базисов; в четвертом и пятом разделах ставится и решается задача разработки энергоэффективных структур «зеленой» логики, в частности, сбоеустойчивых ячеек памяти.

## 1. Анализ логики FPGA в контексте «зеленых» вычислений

Существующая логика FPGA в большей части распределена в пространстве – по кристаллу кремния в виде конфигурируемых логических блоков (КЛБ) [3], состоящих из логических элементов ЛЭ, программируемых локальных и глобальных матриц соединений (ЛМС, ГМС, рис. 1).

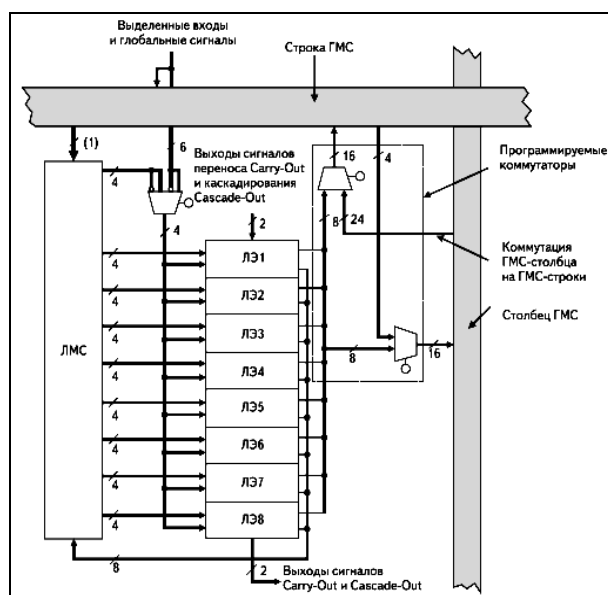


Рис. 1. Конфигурируемый логический блок FPGA

Причем каждый логический элемент ЛЭ строится как постоянное запоминающее устройство ПЗУ (LUT – Look Up Table), представляющее собой мультиплексор 16–1, входы данных которого настраиваются так называемыми конфигурируемыми ячейками памяти (КЯ) (рис. 2) [10]. Элементарные мультиплексоры 2–1 построены на передающих транзисторах (рис. 3).

Подобные структуры используются и в настраиваемых коммутаторах связей. Для компенсации падения напряжения в цепочке передающих транзисторов (имеются ограничения на длину цепочки в соответствии с законами проектирования БИС Мида и Конвея [11]!) должны включаться буферные элементы, которые представляют собой КМДП инверторы.

Интересно, что такие структуры на основе передающих транзисторов – это удивительный возврат на новом уровне к некоему подобию последовательно-параллельных релейно-контактных схем, методы синтеза которых были разработаны в 60-е годы 20-го века профессором Рогинским В.Н. [12].

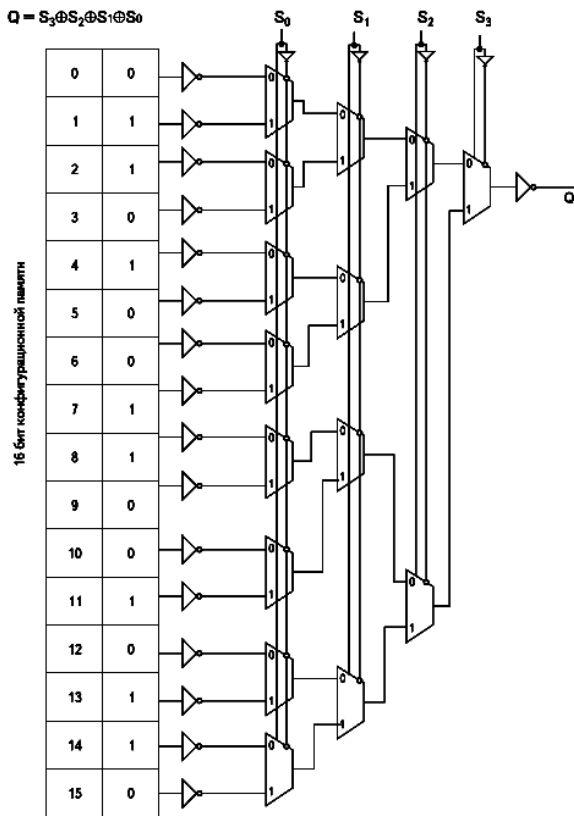


Рис. 2. Логический элемент ПЛИС FPGA

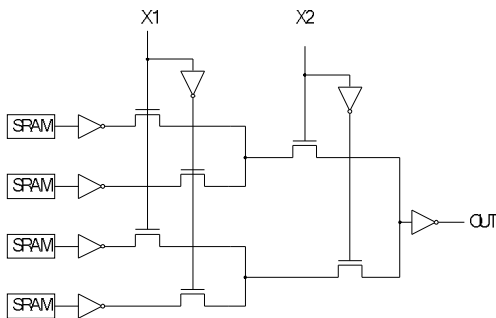


Рис. 3. Схема передающих МОП транзисторов для реализации произвольной функции двух переменных

Настройка дерева мультиплексоров осуществляется путём загрузки так называемой конфигурационной статической оперативной памяти SRAM, шести-транзисторная ячейка которой представлена на рис. 4.

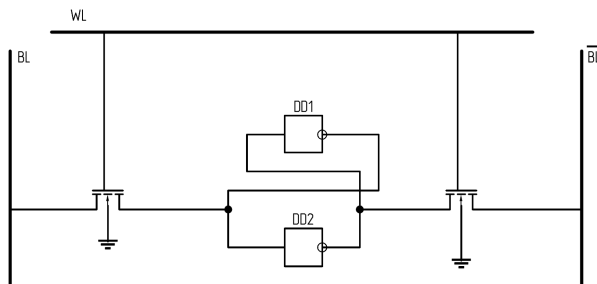


Рис. 4. Ячейка памяти SRAM

Наблюдается устойчивая тенденция увеличения числа входов LUT. Уже разработаны логические элементы с числом переменных 5,6 и даже 7,8.

**Предварительные выводы.** Таким образом, существующая логика FPGA настраивается по функциям и связям соответствующим файлом конфигурации, объём которого составляет миллионы бит, а время загрузки – несколько миллисекунд.

ПЗУ - LUT реализация ЛЭ FPGA основана на совершенных дизъюнктивных нормальных формах (СДНФ) представления логических функций, что требует значительных аппаратных затрат. Так, количество транзисторов в ЛЭ (рис. 2) без учёта коммутаций пропорционально  $2^n$ , где  $n$  – число переменных и определяется выражением:

$$L_{\text{ЛЭ-СДНФ}} = 7 \cdot 2^n + 2n.$$

В самом дереве мультиплексоров на  $n$  переменных их число равно  $2^n - 2, n > 2$ , а в ячейках конфигурации (КЯ) –  $6 \cdot 2^n$ , и ещё  $n+1$  инвертор, в каждом из которых 2 транзистора.

В то же время большая часть реализуемых  $n$  переменных логических функций, например, в управляющих автоматах, имеет число конъюнкций  $k$  много меньшее, чем общее число наборов  $n$  двоичных переменных:

$$k \ll 2^n.$$

Логика FPGA занимает относительно небольшой процент площади кристалла по сравнению с памятью, но от неё во многом зависит энергопотребление и энергоэффективность системы на кристалле. В настоящее время вычисление логических (булевых) функций и автоматных отображений в ЛЭ осуществляется аппаратно, что не позволяет в полной мере реализовать энергосберегающие технологии в FPGA.

## 2. Концепция «зелёной» программируемой логики

**Методы снижения энергопотребления для FPGA проектов** проанализированы в вводной части статьи. Кроме того, в дополнение к ним, обзор таких методов для так называемых energy critical applications (прежде всего, для бортовых систем космических аппаратов с длительным временем активного функционирования) дан в [13].

Методы снижения энергопотребления, базирующиеся на специальных приемах абстрактного и структурного синтеза, декомпозиции конечных автоматов на FPGA, описаны в [14 – 16].

**Основные положения концепции «зелёной» программируемой логики.** Концепция состоит в максимизации энергоэффективности проектов программируемой логики (FPGA проектов) при обеспечении выполнения требований по другим характеристикам (быстродействию, надежности, проектным затратам). Предлагаем ряд положений, которые обобщают и развивают известные подходы, а также содержат новые идеи по энергосбережению для FPGA проектов.

1. Частичное перераспределение логики из пространственной во временную область на основе программно (или, скорее, микропрограммно) - аппаратного вычисления булевых функций или возможной реализации конечных автоматов, точнее, вычисления автоматных отображений. Другими словами, снижение энергопотребления  $E$  (за счёт отключения части аппаратуры  $W$ ) влечёт приемлемое увеличение времени реализации  $T$ :

$$E \downarrow (W \downarrow) \rightarrow T \uparrow.$$

Для этого необходимо многократное использование одного и того же КЛБ в пределах одной загрузки файла конфигурации.

2. Оперативное перераспределение функций (автоматных отображений) между такими конфигурируемыми блоками с целью достижения максимального снижения энергопотребления, при соблюдении заданных временных и прочих ограничений.

3. Оперативная оптимизация каждого конкретного перераспределения для данных условий эксплуатации, с помощью внешних, старших по иерархии систем, перед загрузкой файла конфигурации.

4. Использование для работы логики на пониженном напряжении питания, приводящем к возрастанию интенсивности сбоев, элементов с избыточным базисом (функционально-полных толерантных элементов – ФПТ [17, 18], например, в ячейках конфигурационной памяти). Этот подход может быть распространён на область самосинхронной схемотехники, обеспечивающей работу на ультранизком напряжении КМДП элементов, что было установлено группой Ю.А. Степченкова, ИПИ РАН [19], продолжающих исследования по тематике одного из основоположников теории аперiodических автоматов и их коллективного поведения проф. В.И. Варшавского.

Самосинхронные устройства реализуются, в основном, в базовых матричных кристаллах (БМК). Реализация полностью самосинхронных FPGA проектов нетривиальна и представляет отдельную научно-технологическую проблему. Часто анонсируемые некоторыми фирмами «самосинхронные» FPGA кристаллы, по мнению многих исследователей, являются псевдосамосинхронными решениями.

Для компенсации возрастающей интенсивности сбоев возможно введение структурной избыточности, которая, с одной стороны, уменьшает эту интенсивность для системы, с другой, – увеличивает энергопотребление. Поиск оптимального варианта резервирования по критерию «надёжность-энергопотребление» представляет самостоятельную задачу исследования.

5. Предлагаемые «зелёные», программно-аппаратные логические элементы, по мнению авторов, могут быть эмулированы в существующие FPGA с использованием соответствующих САПР. Однако, может решаться задача разработки новых энергоэффективных архитектур и технологий автоматизиро-

ванного проектирования, специальных настраиваемых компиляторов.

### 3. Программно-аппаратная реализация

**Эволюция решений.** После создания первых микропроцессоров выяснилось, что программная реализация логических функций и автоматов на их базе, которой уделялось огромное внимание в конце 70-х и в начале 80-х годов 20-го века, не даёт решений для ряда практически важных задач. В связи с этим возник интерес к аппаратной поддержке таких вычислений [20 – 22].

В это же время началось активное развитие и программируемых логических устройств (PLD-ПЛУ, сейчас ПЛИС). ПЛУ-ПЛИС заполнили ниши между «жесткой» логикой базовых матричных кристаллов – БМК, программируемых логических матриц – ПЛМ, с одной стороны, и «гибкой» логикой микропроцессоров и микроконтроллеров, – с другой. У авторов свежи воспоминания о статьях Дейва Бёрски и Лизы Малиньяк в журнале «Электроника», которые читались на одном дыхании.

Возникла идея реализации «гибкой» программируемой логической матрицы – ПЛМ, как своего рода логического расширителя, сопроцессора [22]. Однако вскоре усовершенствованные технологии ПЛИС заполнили практически всё «жизненное» пространство и идея была отложена. В ПЛИС же стал доминировать подход на основе гиперизбыточности – реализации логических функций на базе совершенных дизъюнктивных нормальных форм (СДНФ). Для реализации, например, простого инвертора, надо использовать четырёхходовой элемент LUT с доброй сотней транзисторов!

**Логические элементы ПЛИС, ориентированные на дизъюнктивные нормальные формы.** Понимание того, что должны быть и другие подходы, пришло по мере возрастания числа входов и сложности LUT. Поэтому в [23, 24], была предпринята попытка создания логического элемента ПЛИС, ориентированного на менее затратные дизъюнктивные нормальные формы (ДНФ), что является дальнейшим развитием научно технических решений [20 – 22] (рис. 5).

ДНФ-КЛБ содержит блоки конъюнкций БК, блоки значений конъюнкций БЗК и блоки функций БФ. Настройка производится константами  $X_0$ ,  $X_d$ ,  $Z_0$  по числу конъюнкций  $k$  от  $n$  переменных, входящих в  $m$  функций. Тройки констант  $X_0$ ,  $X_d$ ,  $Z_0$  необходимы для каждой из  $k$  конъюнкций в системе из  $m$  функций. Вычисления проводятся по заданному входному вектору  $\bar{X}$ . Такая реализация на основе ФПТ-элементов позволяет получить значительный выигрыш в сложности при реализации функций большого числа переменных.

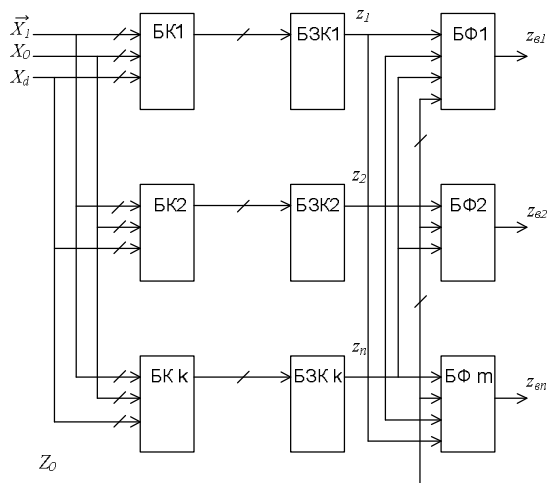


Рис. 5. ДНФ-КЛБ

В то же время создаются условия для эффективного программно-аппаратного вычисления логических функций и автоматных отображений путём многократного использования без обращения к внешней памяти конфигураций.

**Блок программируемой конъюнкции.** Для дальнейшего продвижения такого подхода с использованием МОП передающих транзисторов предложим блок программируемой конъюнкции (матрица «И»), в котором требуемая  $j$ -я конъюнкция  $n$  переменных ( $j = 1, \dots, k$ ) по каждой  $i$ -й переменной будет формироваться с использованием двух бит конфигурационной памяти – конфигурационных ячеек КЯ (рис. 6, 7).

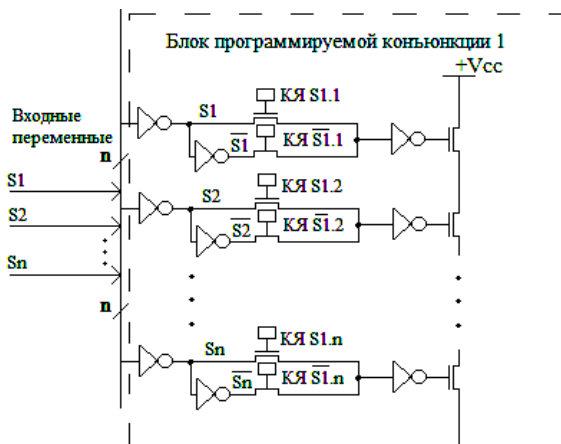


Рис. 6. Блок программируемой конъюнкции логического элемента ПЛИС - FPGA на основе ДНФ реализации логических функций с использованием передающих транзисторов

Оценим аппаратные затраты на реализацию логического элемента ПЛИС - FPGA на основе ДНФ реализации логических функций с использованием передающих транзисторов:

$$L_{\text{ЛЕ-ДНФ}} = 18nk + m(6k + 2).$$

Сравнение по сложности (количеству транзисторов) ЛЭ-СДНФ и ЛЭ-ДНФ представлено на рис. 8. При  $n > 8$  предпочтительна реализация в ДНФ.

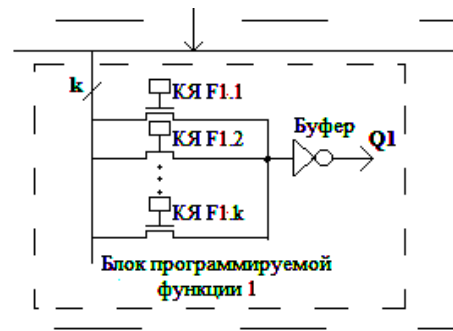


Рис. 7. Блок программируемой дизъюнкции логического элемента ПЛИС - FPGA на основе ДНФ реализации логических функций с использованием передающих транзисторов

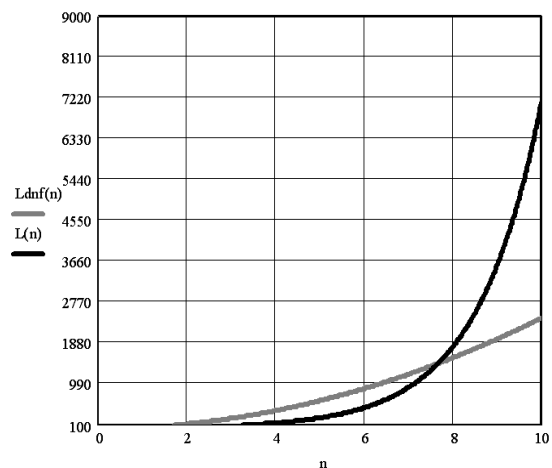


Рис. 8. Сравнение сложности в количестве транзисторов ЛЭ-СДНФ (L) и ЛЭ-ДНФ (Ldnf) при  $n=m=k$

Кроме того, целесообразно исследовать возможность применения в новых условиях всей линейки ранее предложенных устройств аппаратной поддержки вычисления логических функций и автоматов [22 – 25]: от одноконъюнкторного модуля до модулей, обеспечивающих свёртку настроечных масок и пр.

#### 4. Задача перераспределения функций и автоматных отображений между КЛБ для снижения энергопотребления

**Исходные положения.** Ранее перераспределение функций между подсистемами предлагалось с целью поиска оптимальных решений в смысле надёжности [25].

Распространим этот подход на «зелёные» FPGA проекты. Допустим, имеется система  $\Phi = \{\phi_1, \phi_2, \dots, \phi_n\}$  с некоторым множеством конфигурируемых логических блоков.

В этом случае можно считать, что конфигурируемые логические блоки линейно упорядочены, хотя, точнее, они могут быть упорядочены в некоторой матрице – двумерной структуре, в трёхмерном кубе, и, в общем случае, в некотором  $n$ -мерном про-

пространстве. В принципе система  $\Phi$  может нести информацию о любой заранее заданной структуре упорядоченности с соответствующей метрикой. Если задана двумерная матрица, то, зная её размерность, можно по  $\Phi$  получить сведения о нахождении конкретного  $\phi$  в двумерном пространстве.

Исходное энергопотребление  $E(\Phi)$  системы равно сумме энергопотребления отдельных работающих блоков:

$$E(\Phi) = \sum_{i=1}^n \phi_i .$$

Предполагается, что реализация заданных функций, автоматных отображений между конфигурируемыми блоками, осуществляется на основе программно-аппаратных средств, выполняющих часть некритичных функций последовательно во времени. Другими словами, выделяются непрерывно реализуемые функции (НРФ), а остальные, некритичные функции, реализуются периодически.

С этой целью осуществляется некоторое перераспределение  $\chi$  некритичных функций:

$$\Phi_{\chi} = \langle (\phi_i + \xi_i \delta_{ij}) \rangle, i, j = \overline{1, n} ,$$

где  $\xi$  – переменная знака перераспределения ("+" или "-"), которая «принимает» чужие функции для выполнения или «отдаёт»,  $\delta$  – количественный показатель такого перераспределения (сколько и кому).

Интуитивно ясно, что для решения этой задачи необходима своего рода «анизотропность кристалла», то есть наличие неодинаковых свойств – средств программно-аппаратной реализации. Если имеется полная равноценность в смысле энергопотребления – «сколько отняли, столько и приплюсую», то смысла в таком перераспределении нет. Напротив, это приведёт к увеличению энергозатрат, ибо требуется доработка, дополнительные затраты в КЛБ для передачи входных переменных «чужим» блокам, для приёма значений функций, вычисленных «чужими» блоками, для обеспечения работы этих средств, например, регистров при отключенном питании своего блока.

Следовательно, вообще говоря:

$$\Phi_{\chi} = \langle (\phi_i + \Delta\phi + \xi_i \delta_{ij}) \rangle, i, j = \overline{1, n} .$$

Тогда  $E(\Phi_{\chi}) < E(\Phi)$  и перераспределение имеет смысл. Причём параметр  $\delta_{ij}; i, j = \overline{1, n}, i \neq j$ , означает «кто»  $i$ -я «кому»  $j$ -й «отдаёт».

На самом деле «отдача» может быть представлена более сложно: блок может отдавать и не одному блоку, а нескольким, и может, отдавая, принимать, например

$$\Phi_{\varepsilon} = \langle (\phi_i + \Delta\phi + \xi_{ij} \delta_{ij} + \xi_{ik} \delta_{ik} + \dots) \rangle, \\ i, j, k, \dots = \overline{1, n}, i \neq j \neq k \dots$$

**Постановка задачи.** В общем случае задача заключается в поиске  $E(\Phi_{\varepsilon}) \rightarrow \min$  при непревышении ограничений по времени реализации  $t \leq t_{\text{доп}}$ .

К такой постановке задачи может быть добавлена задача снижения напряжения питания, что оправдано для КМДП вентиляей.

Таким образом, при снижении уровня питающего напряжения увеличивается вероятность отказов (сбоев). Для ее компенсации вводится структурная избыточность, которая увеличивает энергопотребление. Нахождение оптимального варианта резервирования, например, градиентным методом, учитывает ограничения по времени вычислений с помощью микропрограммно – аппаратных модулей.

Пусть FPGA включает в свой состав  $n$  подсистем (кластеров) вычисления логических функций. Известны значения вероятности безотказной (бесбойной) работы (ВБР)  $P_i$  для исходного энергопотребления  $W_i$  (где  $i = 1, \dots, n$ ) каждой из подсистем.

Известно, что при снижении энергопотребления  $\Delta W_i$  снижается вероятность  $P_i$  на значение  $\Delta P_i$ . При этом возможно часть функций вычислять микропрограммно – аппаратно. Кроме того, указаны временные ограничения  $T_{\text{доп}}$ . Имеется  $m$  методов повышения вероятности безотказной работы.

Вариант резервирования с использованием микропрограммно – аппаратного вычисления логических функций имеет вид вектора:

$$(k_{i,\eta}); i = \overline{1, n}; \eta = \overline{1, m} .$$

Две постановки задачи оптимизации структурной схемы надёжности (СН) FPGA выглядят следующим образом:

1) Найти  $(k_{i,\eta})$  :

$$W_c \rightarrow \min \text{ при } P_c(t) \geq P_c^{\text{зад}}(t), T \leq T_{\text{доп}} .$$

2) Найти  $(k_{i,\eta})$  :

$$P_c(t) \rightarrow \max \text{ при } W_c \leq W_c^{\text{зад}}, T \leq T_{\text{доп}} ,$$

где  $W_c$  – энергопотребление FPGA проекта,  $P_c(t)$  – вероятность его безотказной (бесбойной) работы.

## 5. Сбоеустойчивая ячейка памяти SRAM с триггером на ФПТ-элементах

**Предлагаемый подход к обеспечению сбоеустойчивости при снижении напряжения питания.** Ограничимся анализом сбоеустойчивости элементов памяти, занимающей большую часть кристалла FPGA.

Модифицируем ячейку SRAM (рис. 4), заменив инверторы избыточными ФПТ-элементами. Сравним предлагаемую схему с троированной ячейкой и мажоритарной схемой на выходе радиационно-стойкой FPGA фирмы Actel [26]. Мажоритирование требует  $4 \times 3 + 2 = 14$  транзисторов (2 – это «боковые» транзисторы записи-считывания).

Схема мажоритирования (реализации функции  $ab \vee ac \vee bc$ ,  $a, b, c$  – выходы трёх ячеек) требует в лучшем случае 12 транзисторов инверсного мажоритарного элемента (подобные имеются в библиотеках БМК) и 2 транзистора для выходного инвертора.

Анализ вариантов. Сравним два варианта реализации триггеров по вероятности бессбойной работы (ВБСР) без учёта «боковых» транзисторов, ВБСР которых предполагается одинаковой для двух вариантов. Зададим  $\lambda$  – интенсивность сбоев транзисторов. Тогда для троированной схемы с одним мажоритарным элементом:

$$P_{\text{ТТ}} = \left( 3e^{-2\cdot 12\lambda t} - 2e^{-3\cdot 12\lambda t} \right) \cdot e^{-14\lambda t},$$

где  $e^{-14\lambda t}$  – ВБСР мажоритарного элемента.

Для триггера на ФПТ элементах с учётом только отказов сбоев не более двух транзисторов (на самом деле парируются и некоторая часть отказов большего числа транзисторов) имеем:

$$P_{\text{фct}}(t) = e^{-16\lambda t} + 16e^{-15\lambda t}(1 - e^{-\lambda t}) + 96e^{-16\lambda t}(1 - e^{-\lambda t})^2.$$

Например, отказ 2-х транзисторов в 4-х частях схемы (в верхней части – подключения “+”, в нижней – подключения “0 вольт” в 2-х инверторах) возможен в  $C_4^2 = 6$  вариантах, т.е.

$$6 \cdot 4 \cdot 4 \cdot e^{-14\lambda t}(1 - e^{-\lambda t})^2.$$

Сравнение ВБСР троированного триггера на мажоритарных элементах ( $P_{\text{tt}}$ ) и триггера на ФПТ элементах ( $P_{\text{fct}}$ ) иллюстрируется рис. 9.

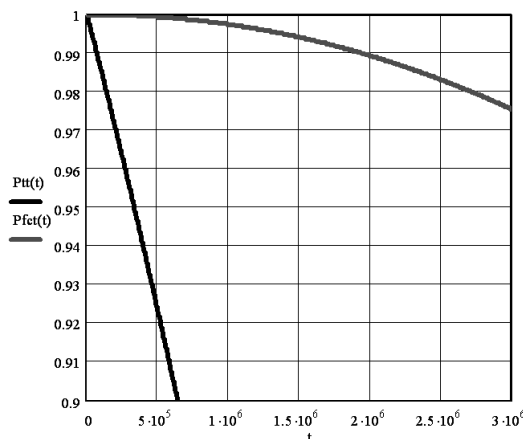


Рис. 9. Сравнительный анализ ВБСР троированного триггера на мажоритарных элементах ( $P_{\text{tt}}$ ) и триггера на ФПТ элементах ( $P_{\text{fct}}$ )

Гистограмма сравнения двух реализаций отказоустойчивых ячеек памяти SRAM (1 – аппаратные затраты в транзисторах, 2 – задержка в транзисторах) показана на рис. 10.

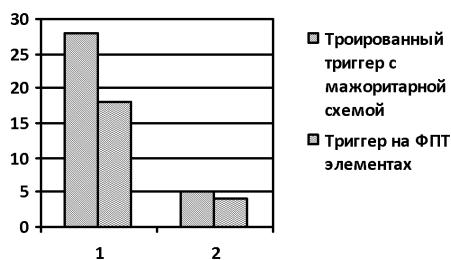


Рис. 10. Гистограмма сравнения двух реализаций отказоустойчивых ячеек памяти SRAM

(1 – аппаратные затраты в транзисторах, 2 – задержка транзисторов)

Таким образом, использование ФПТ-элементов для повышения отказоустойчивости ячейки памяти SRAM является предпочтительным по сравнению с известным вариантом троирования по показателям сложности, быстродействия и вероятности безотказной и бессбойной работы.

## Заключение

**Гонка технологий: энергоэффективная производительность.** Мировые лидеры в области FPGA технологий Aktera, Xilinx, Microsemi и другие ведут «зеленую» гонку как одну из основных составляющих жесткой конкуренции на рынке, сравнимую с такими ее аспектами как надежность, быстродействие, число элементов на кристалле.

Для создания систем на FPGA с высокой энергоэффективной производительностью ЭЭП (производительностью, определяемой не только количеством операций, выполняемых в единицу времени ПВ, а и объемом потребляемой электроэнергии (ЭЭ), ЭЭП = ПВ/ЭЭ) необходимы сбалансированные решения на разных уровнях иерархии: транзисторных ячеек, вентиляных схем, КЛБ, кристаллов, HDL-моделей, модулей и системы в целом.

## Выводы и дальнейшие исследования

В данной работе предложена концепция «зеленой» программируемой логики для FPGA проектов. Сформулирована задача программно-аппаратного перераспределения функций между блоками FPGA, решение которой позволяет найти энергосберегающие варианты реализации логики. Предложены технические решения, обеспечивающие поддержку такого подхода, которые иллюстрируют возможности нахождения компромисса между энергопотреблением и надежностью.

Далее целесообразно исследовать пути увеличения ЭЭП на разных уровнях иерархии с учетом их связанности, возможности выбора и динамической смены базиса в зависимости от отказов КЛБ по критерию энергоэффективности FPGA проектов.

## Список литературы

1. *Green computing* [Электронный ресурс]. – URL: [http://en.wikipedia.org/wiki/Green\\_computing](http://en.wikipedia.org/wiki/Green_computing) (дата обращения 28.12.2012).
2. *Harnessing Green IT: Principles and Practices* / San Murugesan, G.R. Gangadharan (Editors). – Wiley - IEEE. – 2013.
3. *Energy modulated computing* [Электронный ресурс]. – URL: <http://www.cs.ox.ac.uk/seminars/653.html> (дата обращения 31.12.2012).
4. *Green IT Conferences* [Электронный ресурс]. – URL: <http://www.greenitconferences.org/> (дата обращения 31.07.2013).

5. Kharchenko V. Cooperation in Area of Green IT Education and Science: International Projects and University-Industry Partnership / V. Kharchenko, V. Sklyar // *Proceedings of Green IT Conference*. – Moscow, Russia, March, 13, 2013.
6. Kharchenko V. Green Computing and Communications in Critical Application Domains: Challenges and Solutions / V. Kharchenko, A. Gorbenko, V. Sklyar, C. Phillips // *Proceedings of the 10<sup>th</sup> Conferences on Digital Technologies*, May, 29-31, 2013, Žilina, Slovakia. – 2013. – P. 1-8.
7. Anderson J. Low-Power Programmable FPGA Routing Circuitry / J. Anderson // *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. – 2009. – Vol. 17, No. 8. – P. 1048-1060.
8. Volkoviy S. Improving of Technical Characteristics of Systems-on-Programmable Chips Using Internal Circuit Diversity / S. Volkoviy, V. Kharchenko // *Bulletin of Khmelnit-sky National University, Technical Science*. – 2007. – No 2. – P. 153-155.
9. FPGA & SoC Product [Электронный ресурс]. – URL: <http://www.microsemi.com/products/fpga-soc/fpga-and-soc/> (дата обращения 7.1.2013).
10. Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри / С. Цыбин [Электронный ресурс]. – URL: [http://www.kit-e.ru/articles/plis/2010\\_11\\_56.php](http://www.kit-e.ru/articles/plis/2010_11_56.php) (дата обращения 1.1.2013).
11. Правила топологического проектирования Мид-Конвей [Электронный ресурс]. – URL: <http://do.gendocs.ru/docs/index-55156.html> (дата обращения 25.1.2013).
12. Рогинский В.Н. Синтез релейных схем управления / В.Н. Рогинский. – М.: Энергия, 1964. – 424 с.
13. Lamoureux Julien. An Overview of Low-Power Techniques for Field-Programmable Gate Arrays / Lamoureux Julien, Wayne Luk // *IEEE Proceeding NASA/ESA Conference on Adaptive Hardware and Systems, USA*. – 2008. – P.238-245.
14. Sutter Gustavo. FSM Decomposition for Low Power in FPGA / Sutter Gustavo, Todorovich Elias, Lopez-Buedo Boemo, Sergio Eduardo // *Proceedings of the Reconfigurable Computing Is Going Mainstream, 12th International Conference on Field-Programmable Logic and Applications*. – 2002. – P. 350-359.
15. Sambhu Nath Pradhan. Low Power Fnite State Machine Synthesis Using Power-Gating / Sambhu Nath Pradhan, M. Tilak Kumar, Santanu Chattopadhyay // *INTEGRATION, the VLSI journal*. – 2011. – 44 (2011). – P. 175-184.
16. Barkalov A. Hradware Reduction in FPGA-based Moore FSM / A. Barkalov, L. Titarenko, R. Malcheva, K. Soldatov // *Journal of Circuits, Systems and Computers*. – 2013. – Vol 22, 3. – P. 1-20.
17. Kamenskih A.N. Syntesis and Analysis of Self-Timed Functionally Complete Tolerant Element / A.N. Kamenskih, I.S. Ponurovskiy, S.F. Tyurin // *Radioelectronic and Computer Systems*. – 2013. – № 5 (64). – P. 162-167.
18. Tyurin S. Redundant Bases for Critical Systems and Infrastructures General Approach and Variants of Implementation [Text] / S. Tyurin, V. Kharchenko // *Proceedings of the 1st Intrenational Workshop on Critical Infrastructures Safety and Security, Kirovograd, Ukraine 11-13, May, 2011 / V. Kharchenko, V. Tagarev (edits)*. – 2011. – Vol. 2. – P. 300-307.
19. Дьяченко Ю.Г. Система характеристики самосинхронных элементов / Ю.Г. Дьяченко, Н.В. Морозов, Д.Ю. Степченков, Ю.А. Степченков // *Системы и средства информатики*. – 2012. – 22 (1). – С. 38-48.
20. 6T SRAM Cell [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.iue.tuwien.ac.at/phd/entner/node34.html> (дата обращения 7.1.2013).
21. Тюрин С.Ф. Программно-аппаратная реализация логических алгоритмов в микропроцессорных системах / С.Ф. Тюрин, В.С. Харченко, Г.Н. Тимонькин, В.А. Мельников // *Зарубежная радиоэлектроника*. – 1992. – № 2. – С. 24-36.
22. Тюрин С.Ф. Методы аппаратной поддержки логических алгоритмов в микропроцессорных системах / С.Ф. Тюрин, Г.Н. Тимонькин, В.С. Харченко // *Управляющие системы и машины*. – 1993. – № 1. – С. 55-63.
23. Авторское свидетельство (СССР) № 1444892, G11C17/00, G06F7/00. Программируемое логическое устройство / Тюрин С.Ф., Харченко В.С., Ткаченко С.Н., Жихарев В.Я., Улитенко В.П. Опубл. БИ № 46, 1988.
24. Тюрин С.Ф. ДНФ-конфигурируемый логический блок ПЛИС с использованием функционально-полных толерантных элементов / С.Ф. Тюрин, А.В. Набатов // *Радиоэлектронні і комп'ютерні системи*. – 2012. – № 5(57). – С. 37-42.
25. ПЛИС Actel – основа при реализации "SoC" бортовой аппаратуры. [Электронный ресурс]. – URL: <http://www.spigl.wordpress.com/2009/09/16/pliuc/> (дата обращения 13.1.2013).
26. Тюрин С.Ф. Перераспределение функций между подсистемами с целью получения системного эффекта / С.Ф. Тюрин, А.В. Силин, А.Ю. Беляков, А.В. Яковлев // *Теоретические и прикладные аспекты промышленной автоматизации и информатизации общества: Сб. научн. тр. Пермского ГНИИУМС*. – Пермь, 1995. – Вып. 44. – С. 132-141.

Поступила в редколлегию 1.08.2013

Рецензент: д-р техн. наук, проф. А.В. Дрозд, Одесский национальный политехнический университет, Одесса.

### «ЗЕЛЕНА» ПРОГРАМОВАНА ЛОГІКА: КОНЦЕПЦІЯ ТА ЕЛЕМЕНТИ РЕАЛІЗАЦІЇ ДЛЯ FPGA ПРОЄКТІВ

С.Ф. Тюрін, В.С. Харченко

Аналізуються методи енергозбереження для проєктів програмованої логіки на FPGA кристалах. Пропонуються концепція «зеленої» програмованої логіки для FPGA проєктів та елементи її реалізації на ненадмірних базисах і надмірних базисах, толерантних до відмов. Досліджуються варіанти створення енергоєфективної логіки FPGA на основі програмно-апаратної реалізації бульових функцій та автоматів і перерозподілу функцій між блоками таких реалізацій з урахуванням необхідності толерування збоїв при зменшенні напруги живлення.

**Ключові слова:** зелена логіка, програмована логічна інтегральна схема, FPGA, конфігурований логічний блок, логічний елемент, Look Up Table – LUT, збоєстійкість.

### «GREEN» PROGRAMMABLE LOGIC: A CONCEPTION AND IMPLEMENTATION EXAMPLES FOR FPGA DESIGNS

S.F. Tyurin, V.S. Kharchenko

The methods of power saving for FPGA designs are analysed. A conception of "green" programmable logic for FPGA designs and an examples of its implementation by use of irredundant and redundant fault-tolerant basis are suggested.



*Techniques of development power effective FPGA-oriented logic are researched. These techniques are based on the SW-HW co-design of Boolean functions and automata and functions reallocation between logic blocks for such implementations taking into account requirements to tolerating of soft faults caused by reducing of power consumption.*

**Keywords:** *green programmable logic, FPGA, configurable logic block, logic element, Look Up Table – LUT, fault-tolerance.*