

УДК 681.32

# Комбинированное резервирование энергоэффективной программируемой логики для критических областей применения

А. С. Прохоров<sup>1</sup>, С. Ф. Тюрин<sup>1,2</sup>

<sup>1</sup>Пермский национальный исследовательский политехнический университет  
Россия, 614990, Пермь, ул. Комсомольский пр., 29

<sup>2</sup>Пермский государственный национальный исследовательский университет  
Россия, 614990, Пермь, ул. Букирева, 15  
tyurinsergfeoyandex.ru; +7-952-32-02-510

Получены выражения, описывающие комбинированное резервирование программируемой логики для энергоэффективных, но критических областей применения, например для аппаратуры космических аппаратов. Анализируются варианты реализации резервирования на транзисторном уровне и резервирования каналов. Показано, что оптимальные решения лежат в области комбинирования различных вариантов и могут обеспечить достижение минимума энергопотребления при заданной надежности.

**Ключевые слова:** *программируемая логика; отказо- и сбоеустойчивость; логическая функция.*

DOI: 10.17072/1993-0550-2017-3-82-86

## Введение

В настоящее время активно развивается направление энергоэффективной электроники "Грин Компьютинг" (Green Computing, Green Co) [1–3]. Широко используются разнообразные режимы "сна" и так называемое динамическое масштабирование частоты и напряжения питания, которое сейчас может быть снижено до 0,8 вольт! [4–6]. В том числе этот тренд наблюдается и в различных областях программируемой логики [7–11]. Такие энергоэкономные устройства востребованы в различных областях, однако при снижении напряжения питания резко возрастает интенсивность сбоев, что негативно сказывается на работоспособности аппаратуры, например, работающей на космических аппаратах в условиях воздействия космической радиации, тяжелых заряженных частиц [12]. Таким образом, с одной стороны, требуется снижать напряжение питания, а с другой – необходимо обеспечить заданный уровень надежности. И в то же время при создании отказо- и сбое-

устойчивой аппаратуры необходимо введение избыточности (резервирование), что, в свою очередь, повышает энергопотребление. Предлагается подход комбинирования избыточности различного уровня с целью обеспечения минимального энергопотребления для заданного уровня надежности программируемой логики. В дальнейшем отказо- и сбоеустойчивость будем называть сокращенно *отказоустойчивостью*, помятуя, о том что сбой – это кратковременный, самоустраниющийся отказ [13].

## 1. Отказоустойчивая FPGA-реализация логических функций, заданных в СДНФ

Логическая функция  $n$  переменных  $x$ , реализуемая на выходе дерева передающих транзисторов LUT (Look Up Table) ПЛИС FPGA, [7, 8] может быть представлена следующим образом:

$$\bar{z}_{out} = \bigvee_{i=1}^{2^n} \left( \bigwedge_{j=1}^n x_j^{\sigma(i-1,j)} \cdot \bar{d}_i \right), \quad (1)$$

где  $\sigma(i-1, j) \in \{0, 1\}$  равно значению  $j$ -го разряда в двоичной записи числа  $i-1$ ,  $d_i \in \{0, 1\}$  – значение  $i$ -й конфигурационной ячейки статической оперативной памяти SRAM.

Парирование  $r$  отказов по некоторой  $i$ -й переменной предполагает выражение

$$\bar{z}_{out.ft} = \bigvee_{i=1}^{2^n} \left( \bigwedge_{j=1}^n \bigwedge_{\chi=1}^{r+1} (\& x_{i,\chi}) \right)^{\sigma(i-1,j)} \cdot \bar{d}_i. \quad (2)$$

При этом затраты по каждой переменной в каждой ветви дерева возрастают с 1 до  $(r+1)^2$ . Вероятность безотказной работы транзисторов  $P(t)$  по каждой переменной для экспоненциальной модели (модели Вейбулла) [14] имеет вид

$$P(t) = \sum_{i=0}^r C_{(r+1)^2}^i \{ e^{-(r+1)^2 - i} \cdot \lambda \cdot t^\alpha \cdot (1 - e^{-\lambda t^\alpha})^i \}, \quad (3)$$

где  $\lambda$  – интенсивность отказов одного канала,  $\alpha$  – коэффициент распределения Вейбулла;  $1 < \alpha < 2$ ,  $t$  – время работы,  $r$  – число парированных отказов. На современном уровне технологии, как правило,  $q=4$ . То есть, требуется декомпозиция дерева (2) передающих транзисторов по  $q$ . При декомпозиции сложного дерева по  $k$ -LUT,  $k \in \{1, 2, 3, 4, \dots, d\}$ ,  $n > k$ :

$$L_{n,k} = 2^{\lfloor n \rfloor} \cdot 8 + (2^{\lfloor k \rfloor + 1} + 6 \lfloor k \rfloor) \cdot \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor} + \quad (4)$$

$$+ (2^{\lfloor n \rfloor - \lfloor \frac{n}{k} \rfloor \cdot \lfloor k \rfloor} + 6 \cdot \left( \lfloor n \rfloor - \lfloor \frac{n}{k} \rfloor \cdot \lfloor k \rfloor \right) + 6 \lfloor n \rfloor),$$

где  $2^{\lfloor k \rfloor + 1} + 6 \lfloor k \rfloor$  – сложность одного  $k$ -дерева  $\lfloor \dots \rfloor$  – округление в нижнюю сторону (floor); таких деревьев ( $k$ -LUT) необходимо в первом слое  $2^{\lfloor n \rfloor - \lfloor k \rfloor}$ , затем нужно провести декомпозицию  $k$ -LUT этого первого слоя, получаем  $2^{\lfloor n \rfloor - \lfloor k \rfloor - \lfloor k \rfloor}$ . Всего необходимо  $i$   $k$ -LUT, где  $i$

определяется из соотношения:  $\lfloor i \rfloor = \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor$  всего

$\sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} 2^{\lfloor n \rfloor - \lfloor i \rfloor \cdot \lfloor k \rfloor}$  и последний LUT на  $\lfloor n \rfloor - \left\lfloor \frac{\lfloor n \rfloor}{\lfloor k \rfloor} \right\rfloor \cdot \lfloor k \rfloor$  переменных.

## 2. Дубль-LUT и LUT с контролем

Неактивное поддерево LUT можно использовать для вычисления второй логической функции. Для этого необходимо дублировать старшую переменную, ввести, напри-

мер, переменную  $x_n^*$  и дублировать настройки, задавая второй набор данных  $d$ . Получим дубль-LUT.

Таблица истинности  $\bar{d}_{2,i}$  записывается старшими разрядами вперед, чтобы использовать неактивную часть дерева. Управление подключением настройки каждой половины дерева осуществляется переменной  $t(x_n)$ :

$$\bar{z}_{1.out.x^n} = \bigvee_{i=1}^{2^{n-1}} \left( \bigwedge_{j=1}^{n-1} x_j^{\sigma(i-1,j)} \wedge \bar{d}_{1,i} \right) \wedge t(x_n) \vee \quad (5)$$

$$\bigvee_{i=2^{n-1}}^{2^n} \left( \bigwedge_{j=1}^{n-1} x_j^{\sigma(i-1,j)} \wedge \bar{d}_{1,i} \right) \wedge \bar{t}(x_n),$$

$$\bar{z}_{2.out.(x^*)^n} = \bigvee_{i=1}^{2^{n-1}} \left( \bigwedge_{j=1}^{n-1} x_j^{\sigma(i-1,j)} \right) \bigwedge_{i=2^{n-1}}^{2^n} \bar{d}_{2,i} \wedge \bar{t}(x_n) \vee$$

$$\bigvee_{i=2^{n-1}}^{2^n} \left( \bigwedge_{j=1}^{n-1} x_j^{\sigma(i-1,j)} \right) \bigwedge_{i=2^n}^{2^{n-1}} \bar{d}_{2,i} \wedge t(x_n).$$

Предложенный подход можно использовать для контроля вычислений, например в случае работы в условиях воздействия ТЗЧ [12], тогда функция ошибки имеет вид

$$f = \bar{z}_{1.out.x^n} \oplus \bar{z}_{2.out.(x^*)^n}. \quad (6)$$

Соответствующая схема имеет вид (рис. 1)

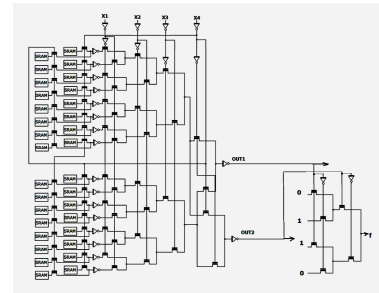


Рис. 1. Дубль-LUT на 4 переменные, вычисляющий две функции одновременно с возможностью контроля правильности вычислений

Отказоустойчивая реализация Дубль-LUT с контролем имеет вид:

$$\bar{z}_{1.out.x^n} = \bigvee_{i=1}^{2^{n-1}} \left( \bigwedge_{j=1}^{n-1} (\& x_{i,j}) \right)^{\sigma(i-1,j)} \bigwedge_{\chi=1}^{r+1} \overline{\bigvee_{\chi=1}^{r+1} \& d(x_n)_{i,\chi}} \bigwedge_{\chi=1}^{r+1} \overline{\bigvee_{\chi=1}^{r+1} \& t(x_n)_{i,\chi}} \vee$$

$$\bigvee_{i=2^{n-1}}^{2^n} \left( \bigwedge_{j=1}^{n-1} (\& x_{i,j}) \right)^{\sigma(i-1,j)} \bigwedge_{\chi=1}^{r+1} \overline{\bigvee_{\chi=1}^{r+1} \& d(x_n)_{i,\chi}} \bigwedge_{\chi=1}^{r+1} \overline{\bigvee_{\chi=1}^{r+1} \& t(x_n)_{i,\chi}},$$

$$\bar{z}_{2.out.(x^*)^n} = \bigvee_{i=1}^{2^{n-1}} \left( \bigwedge_{j=1}^{n-1} (\& x_{i,j}) \right)^{\sigma(i-1,j)} \bigwedge_{i=2^{n-1}}^{2^n} \overline{\bigvee_{\chi=1}^{r+1} \& d(x_n)_{i,\chi}} \bigwedge_{\chi=1}^{r+1} \overline{\bigvee_{\chi=1}^{r+1} \& t(x_n)_{i,\chi}} \vee$$

$$\bigvee_{i=2^{n-1}}^{2^n} \left( \bigwedge_{j=1}^{n-1} (\& x_{i,j}) \right)^{\sigma(i-1,j)} \bigwedge_{i=2^{n-1}}^{2^n} \overline{\bigvee_{\chi=1}^{r+1} \& d(x_n)_{i,\chi}} \bigwedge_{\chi=1}^{r+1} \overline{\bigvee_{\chi=1}^{r+1} \& t(x_n)_{i,\chi}},$$

$$f = \overline{\bigvee_{\chi=1}^{r+1} (\& z_\chi)} \oplus \overline{\bigvee_{\chi=1}^{r+1} (\& z_\chi)}. \quad (7)$$

В случае использования (7), получаем дублированную схему с транзисторным резервированием в каждом канале и в схеме

сравнения. В качестве варианта предлагается резервирование только в схеме сравнения.

### 3. Отказоустойчивая реализация программируемых систем логических функций

DC LUT-реализация систем логических функций [15] в СДНФ:

$$d_{out,i} = \&(x_j^{\sigma(i-1,j)} \vee x_j^{\bar{\sigma}(i-1,j)}); i = 1, 2^n. \quad (8)$$

Отказоустойчивая реализация с учетом (8) имеет вид

$$d_{out,i} = \&([\vee_{j=1}^n (\& x_{i,\chi})]_{\chi=1}^{\sigma(i-1,j)} \vee [\vee_{j=1}^{r+1} (\& x_{i,\chi})]_{\chi=1}^{\bar{\sigma}(i-1,j)}); i = 1, 2^n. \quad (9)$$

Для программирования значений  $m$  логических функций используем

$$z_\theta = \vee_{i=1}^{2^n} ([\vee_{\chi=1}^{r+1} (\& d_{i,\chi})]_{out,i} \cdot [\vee_{\chi=1}^{r+1} (\& h_{i,\chi})]_{li}); 1 = 1, m; \quad (10)$$

где  $h$  – настройка вхождения конstituенты  $i$  в данную функцию  $z_\theta$  из  $m$  функций системы.

В случае реализации систем логических функций в ДНФ (в программируемых логических матрицах – ПЛИМ и в CPLD и др.) [10, 11, 16] ДНФ-парирование  $r$  отказов по некоторой  $i$ -й переменной в подматрице и описывается выражением

$$f_j = \&\{[\vee_{\chi=1}^n (\& x_{i,\chi})] \vee [\vee_{\chi=1}^{r+1} (\& s_{i,\chi})] \cdot [\vee_{\chi=1}^{r+1} (\& x_{i,\chi})] \vee [\vee_{\chi=1}^{r+1} (\& s_{i,\chi})]\}; j = 1, k. \quad (11)$$

Для подматрицы ИЛИ на  $m$  функций с настройкой  $g$ :

$$z_\zeta = \vee_{i=1}^m [\& g_{i,\chi}]_i; \zeta = 1..m; j = 1..k. \quad (12)$$

### 4. Комбинированное резервирование

Рассмотренное выше резервирование представляет собой крайний случай, когда резервируются сами транзисторы. Другой крайний случай – резервирование каналов аппаратуры. Например, троирование (мажоритирование 2 из 3) применяется, например, в ПЛИС Virtex фирмы Xilinx [17–19]. Троированный LUT2 изображен на рис. 2.

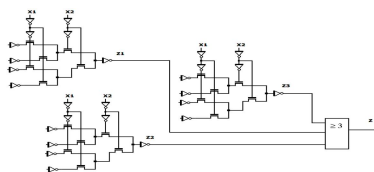


Рис. 2. Троированный LUT2

Троирование DC LUT имеет вид:

$$\begin{aligned} d_{out,i,1} &= \&(x_{j,1}^{\sigma(i-1,j)} \vee x_{j,1}^{\bar{\sigma}(i-1,j)}); \\ d_{out,i,2} &= \&(x_{j,2}^{\sigma(i-1,j)} \vee x_{j,2}^{\bar{\sigma}(i-1,j)}); \\ d_{out,i,3} &= \&(x_{j,3}^{\sigma(i-1,j)} \vee x_{j,3}^{\bar{\sigma}(i-1,j)}); \\ d_{out,i} &= (d_{out,i,1} \wedge d_{out,i,2}) \vee (d_{out,i,2} \wedge d_{out,i,3}) \vee (d_{out,i,1} \wedge d_{out,i,3}); \\ i &= 1..2^n. \end{aligned} \quad (13)$$

В случае троирования мажоритаров получим:

$$\begin{aligned} z_{out,\xi} &= (\bar{z}_{out1} \wedge \bar{z}_{out2}) \vee (\bar{z}_{out2} \wedge \bar{z}_{out3}) \vee (\bar{z}_{out1} \wedge \bar{z}_{out3}), \xi = 1, 2, 3; \\ d_{1,out,i} &= (d_{1,out,i} \wedge d_{2,out,i}) \vee (d_{2,out,i} \wedge d_{3,out,i}) \vee (d_{1,out,i} \wedge d_{3,out,i}); \\ i &= 1..2^n. \end{aligned} \quad (14)$$

Предлагаемое комбинированное резервирование заключается в поиске структуры, в которой используется как резервирование каналов (например, троирование), так и резервирование на уровне транзисторов. Можно показать, что в ряде случаев при этом возможно и резервирование источника питания, а также резервирование связей. Дело в том, что, как показывает моделирование, резервирование каналов может привести к увеличению энергопотребления по сравнению с резервированием на уровне транзисторов. В то же время избыточность такого резервирования, как правило, превышает поканальное резервирование. Зато оно дает значительный эффект увеличения надежности (рис. 3).

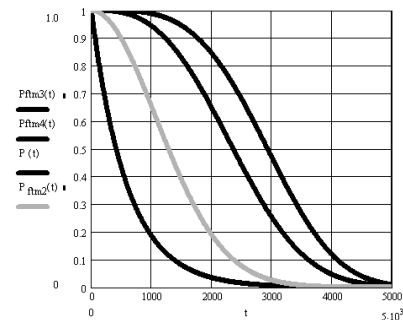


Рис. 3. Изменение вероятности безотказной (бессбойной) работы элементарной схемы FCT, парирующей один отказ  $P(t)_{fm2}$ , схемы FCT, парирующей два отказа  $P(t)_{fm3}$  и схемы FCT, парирующей три отказа  $P(t)_{fm4}$  при интенсивности отказов 10 в минус пятой степени 1/час

Отметим также, что было выявлено промежуточное между двумя крайними случаями – "глубокое" мажоритирование. Кроме

того, имеет смысл резервировать на транзисторном уровне сами устройства для голосования. Поиск оптимальной структуры может быть осуществлен градиентным методом по показателю "вероятность/энерго-потребление", что показано в [20]. В случае допустимости активной отказоустойчивости следует вместо вероятности безотказной работы использовать показатель достоверности функционирования.

## Выводы

Расчеты показывают, что резервирование на транзисторном уровне по вероятности безотказной (бессбойной) работы превосходит резервирование на уровне каналов. Однако оно наталкивается на ограничения Мида–Конвей [21], что приводит к необходимости декомпозиции, кроме того, значительно увеличивается задержка схемы, имеются проблемы с резервированием источника питания и связей.

Оптимизация, на наш взгляд, заключается в поиске "золотой середины" резервирования, лежащей между резервированием каналов и резервированием транзисторов. Установлено, что все виды резервирования ухудшают показатели быстродействия, поэтому целесообразно учитывать и это ограничение.

## Список литературы

1. *Kharchenko V., Kondratenko Y., Kasprzyk J.* (Eds.): Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control, Vol. 74. Berlin, Heidelberg: Springer International Publishing, (2017), DOI: 10.1007/978-3-319-44162-7.
2. *Tyurin S., Kamenskih A.* Green Logic: Models, Methods, Algorithms. In book: Green IT Engineering: Concepts, Models, Complex Systems Architectures, Studies in Systems, Decision and Control, V. Kharchenko, Y. Kondratenko, J. Kasprzyk (Eds.), Vol. 74. Berlin, Heidelberg: Springer International Publishing, 69–86 (2017), DOI: 10.1007/978-3-319-44162-7\_13.
3. *Julien Lamoureux and Wayne Luk.* An Overview of Low-Power Techniques for Field-Programmable Gate Arrays. URL: <http://www.doc.ic.ac.uk/~wl/papers/08/ahs08jl.pdf> (дата обращения: 04.07.2017).
4. *Varghese George, Hui Zhang, Jan Rabaey.* The Design of a Low Energy FPGA. URL: <https://courses.cs.washington.edu/courses/cse591n/07au/papers/p116.pdf> (дата обращения: 09.12.2016).
5. *Fred Wickersham.* Design techniques for FPGA power optimization. URL: <http://signal-processing.mil-embedded.com/articles/design-techniques-fpga-power-optimization/> (дата обращения: 13.03.2017).
6. *Mehta, Nikil.* An ultra-low-energy, variation-tolerant FPGA architecture using component-specific mapping. Dissertation (Ph.D.), California Institute of Technology. (2013) Available at: URL: <http://thesis.library.caltech.edu/7226/1/Nikil-Mehta-2013.pdf> (accessed: 09.12.2016).
7. *Угрюмов Е.П.* Цифровая схемотехника: учебное пособие. СПб.: 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2007. 782 с.
8. *А. Строгонов, С. Цыбин.* Программируемая коммутация ПЛИС: взгляд изнутри. URL: [http://www.kite.ru/articles/plis/2010\\_11\\_56.php](http://www.kite.ru/articles/plis/2010_11_56.php) (дата обращения: 13.03.2017).
9. *Золотуха Р., Комолов Д.* Stratix III – новое семейство FPGA фирмы Altera. URL: [http://kit-e.ru/assets/files/pdf/2006\\_12\\_30.pdf](http://kit-e.ru/assets/files/pdf/2006_12_30.pdf) (дата обращения: 14.03.2017).
10. *Виды программируемой логики.* URL: <http://www.pvsm.ru/programmirovanie/87810> (дата обращения: 10.06.2017).
11. *Programmable Logic Devices.* URL: [http://ee.sharif.edu/~logic\\_circuits\\_t/readings/PLD.pdf](http://ee.sharif.edu/~logic_circuits_t/readings/PLD.pdf) (дата обращения: 04.06.2017).
12. *Austin James Womac.* The Characterization of a CMOS Radiation Hardened-by-Design Circuit Technique. URL: [http://trace.tennessee.edu/cgi/viewcontent.cgi?article=2832&context=utk\\_gradthes](http://trace.tennessee.edu/cgi/viewcontent.cgi?article=2832&context=utk_gradthes) (дата обращения: 04.07.2017).
13. *ГОСТ 27.002-89.* Надежность в технике. Основные понятия. Термины и определения. М.: Изд-во стандартов, 1990. 42 с.
14. *Надежность и эффективность в технике: справочник: в 10 т. / ред. совет во главе с В.С. Авдеевским (предс.) [и др.]. Т.2: Математические методы в теории надежности и эффективности / под ред. Б.В. Гнеденко. М.: Машиностроение, 1987. 280 с.*
15. *Тюрин С.Ф., Вихорев П.В.* Программируемое логическое устройство. Патент РФ № 2573732. Оpubл. БИ № 3 27.01.2016.
16. *CPLD (Complex Programmable Logic Device).* URL: <http://www.myshared.ru/slide/981511/> (дата обращения: 09.06.2017).

17. *Carl Carmichael*. Triple Module Redundancy Design Techniques for Virtex FPGAs URL: [https://www.xilinx.com/support/documentation/application\\_notes/xapp197.pdf](https://www.xilinx.com/support/documentation/application_notes/xapp197.pdf) (дата обращения: 07.06.2017).
18. *Xilinx* Reduces Risk and Increases Efficiency for IEC61508 and ISO26262 Certified Safety Applications. WP461 (v1.0) April 9, 2015. URL: [http://www.xilinx.com/support/documentation/white\\_papers/wp461functional-safety.pdf](http://www.xilinx.com/support/documentation/white_papers/wp461functional-safety.pdf) (дата обращения: 20.12.2016).
19. *QPro* Virtex-II 1.5V Platform FPGAs. DS122 (v3.0) April 7, 2014. URL: [http://www.xilinx.com/support/documentation/data\\_sheets/ds122.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds122.pdf) (дата обращения: 20.12.2016).
20. *Kamenskih, A.N., Tyurin*. The development of basic logic element for strictly self-timed FPGA. International Conference on Information and Digital Technologies, IDT 2015. P. 128–131. DOI: 10.1109/DT.2015.7222962.
21. *Ульман Дж. Д.* Вычислительные аспекты СБИС / пер. с англ. А.В. Неймана / под ред. П.П. Пархоменко. М.: Радио и связь, 1990. 480 с.

## Combined redundancy of the energy-efficient programmable logic for critical applications

A. S. Prokhorov<sup>1</sup>, S. F. Tyurin<sup>1,2</sup>

<sup>1</sup>Perm National Research Polytechnic University; 29, Komsomolsky prospekt, Perm, 614990, Russia

<sup>2</sup>Perm State University; 15, Bukireva st., Perm, 614990, Russia

tyurinsergfe@yandex.ru; +7 952-320-02-510

Expressions are obtained describing the combined redundancy of programmable logic for energy-efficient but critical applications, for example, for spacecraft apparatus. The paper analyzes options for implementing redundancy at the transistor level and channel redundancy. It is shown that the optimal solutions lie in the field of combining different options and can ensure the achievement of the minimum energy consumption for a given reliability.

**Keywords:** *programmable logic; fault tolerance; logic function.*