

УДК 681.3

С.Ф. ТЮРИН¹, А.В. НАБАТОВ²¹ Пермский национальный исследовательский политехнический университет, Россия² Пермский военный институт внутренних войск МВД РФ, Россия

ДНФ-КОНФИГУРИРУЕМЫЙ ЛОГИЧЕСКИЙ БЛОК ПЛИС С ИСПОЛЬЗОВАНИЕМ ФУНКЦИОНАЛЬНО-ПОЛНЫХ ТОЛЕРАНТНЫХ ЭЛЕМЕНТОВ

В статье предлагается и анализируется конфигурируемый логический блок (КЛБ), реализующий логические функции в ДНФ (дизъюнктивной нормальной форме) – ДНФ КЛБ. За основу взят аппаратно реализованный алгоритм программируемой логической матрицы (АПЛА), причём в качестве базисного элемента используется функционально-полный толерантный (ФПТ) элемент, что позволяет получить существенный выигрыш в количестве элементов по сравнению с эквивалентным по числу транзисторов базисом 4И-НЕ и тем более, 2И-НЕ. Предложены процедуры парирования отказов с возможным переходом к микропрограммно-аппаратной реализации логики ПЛИС.

Ключевые слова: программируемая логическая интегральная схема (ПЛИС), конфигурируемый логический блок (КЛБ), совершенная дизъюнктивная нормальная форма (СДНФ), дизъюнктивная нормальная форма (ДНФ), отказоустойчивость, функционально-полный толерантный (ФПТ) элемент, аппаратно реализованный алгоритм программируемой логической матрицы (АПЛА).

Введение

В настоящее время так называемые «крупнозернистые» конфигурируемые логические блоки (КЛБ) программируемых логических интегральных схем (ПЛИС) строятся на основе программируемых ПЗУ – LUT (Look Up Tables) [1]. Эта реализация основана на совершенных дизъюнктивных нормальных формах (СДНФ) представления логических функций, что требует значительных аппаратных затрат.

В то же время большая часть реализуемых логических функций имеет число конъюнкций k много меньше, чем общее число наборов n двоичных переменных:

$$k \ll 2^n. \quad (1)$$

Предлагается ДНФ –реализация логики на основе аппаратно реализованного алгоритма ПЛА [2..5] –АПЛА с использованием в качестве базисных функционально-полных толерантных (ФПТ) элементов [6, 7].

При этом, помимо значительного сокращения аппаратных затрат, создаются условия восстановления логики программируемых логических интегральных схем даже при наличии отказа в каждом конфигурируемом логическом блоке – за несколько тактов, что, конечно, снижает быстродействие, но позволяет ПЛИС реализовать хотя бы часть наиболее важных алгоритмов, допускающих снижение производительности.

1. Аппаратная реализация алгоритма ПЛА –АПЛА на основе ДНФ-КЛБ

Предлагаемый ДНФ - КЛБ по алгоритму АПЛА[8] является дальнейшим развитием научно технических решений [9,10] и представлен на рис. 1.

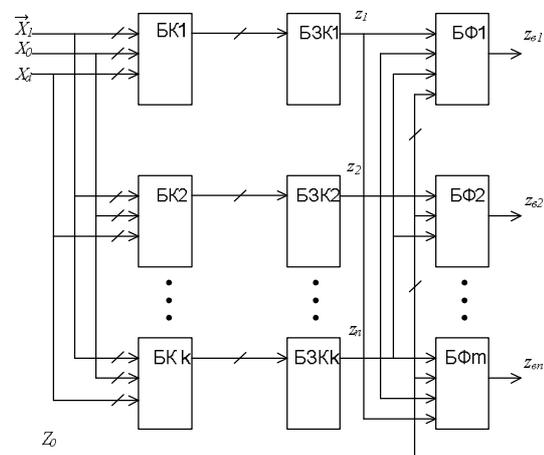


Рис. 1. ДНФ-КЛБ по алгоритму АПЛА

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. То есть 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} .

Блоки конъюнкций БК совместно с блоками значений конъюнкций БЗК вычисляют значения конъюнкций. Если конъюнкция истинна, то есть равна единице, то в блоках функций БФ активируются соответствующие маске Z_0 для данной конъюнкции разряды. Блоки вычисления функций БФ предназначены для вычисления соответствующей j -й выходной функции

$$Z_{vj} = \bar{z}_1 z_{10j} \vee \bar{z}_2 z_{20j} \vee \dots \vee \bar{z}_k z_{k0j} \quad (2)$$

по результатам вычислений в блоках значений конъюнкций БК, когда результат вычисления соответствующей i -й конъюнкции в i -м блоке значения конъюнкций БЗК равен нулю

$$(z_i = 0) . \quad (3)$$

(конъюнкция равна единице) и эта конъюнкция входит в соответствующую функцию системы логических функций

$$(z_{i0j} = 1) . \quad (4)$$

В каждом блоке конъюнкций БК вычисляется n поразрядных функций

$$z_i^* = (x_i x_{i0}) \oplus x_{id} = \overline{x_i x_{i0} x_{id}} \vee x_i x_{i0} \bar{x}_{id} \quad (5)$$

выделения существенных переменных путём конъюнкции с соответствующим разрядом основной маски (o) значащих бит во входном векторе, и суммирования по модулю два значения этого i -го бита, если он значащий, с дополнительной маской (d).

В каждом i -м блоке значений конъюнкций БЗК вычисляется функция:

$$z_i = \bar{z}_1 \bar{z}_2 \bar{z}_3 \bar{z}_4 \bar{z}_5 \bar{z}_6 \bar{z}_7 \dots \bar{z}_n \quad (6)$$

равная единице, когда по всем разрядам z_i данной n -разрядной конъюнкции получены «нули», то есть соответствующая конъюнкция покрывает входной вектор (равна единице или истинна).

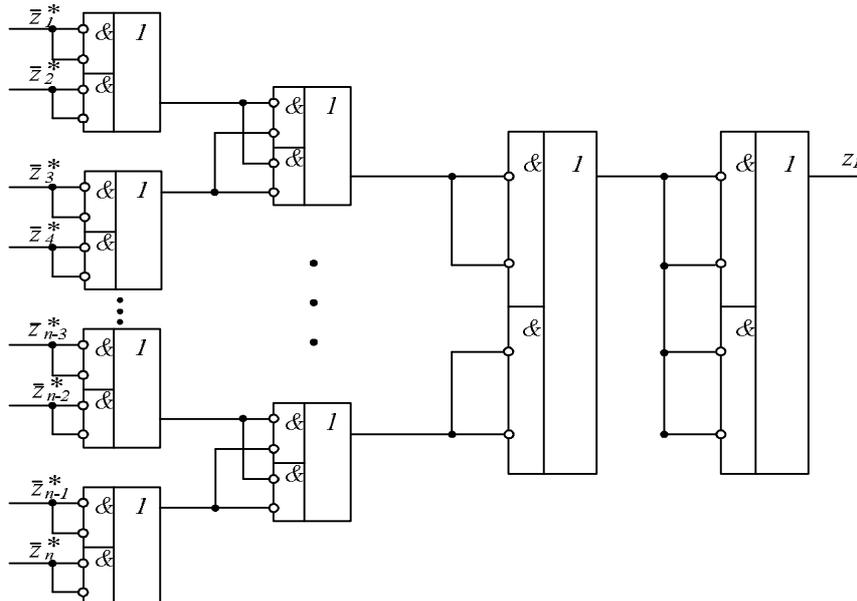


Рис. 3. Блок значений конъюнкций –БЗК

2. Реализация блока конъюнкций -БК

Представление одного из n разрядов блока конъюнкций БК(Рис.2)в ФПТ базе [6,7] имеет вид:

$$z_i = \overline{x_i x_{i0} x_{id}} \vee \overline{x_i} \vee \overline{x_{i0} x_{id}} . \quad (7)$$

Такое представление требует 3-х ФПТ элементов с базисной функцией $\bar{x}_1 \bar{x}_2 \vee \bar{x}_3 \bar{x}_4$ при условии наличия парафазных входов, либо 6 при отсутствии таковых (для получения инверсий x_i, x_{i0}, x_{id}). Реализация в базе И-НЕ требует 4-х элементов при условии наличия парафазных входов, либо 7 при их отсутствии. При этом задержка в обоих случаях равна двум элементам.

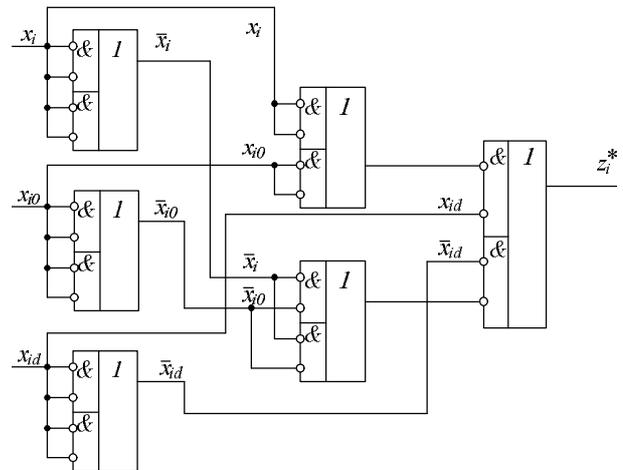


Рис. 2. Блок конъюнкции БК

3. Реализация блока значений конъюнкций

Реализация блока значений конъюнкций БЗК в ФПТ базе представлена на рис.3.

На выходе БК реализуется функция:

$$z = (z_1 \vee z_2) \cdot (z_3 \vee z_4) \vee \dots \vee (z_{n-3} \vee z_{n-2}) \cdot (z_{n-1} \vee z_n), \quad (8)$$

равная единице, если данная конъюнкция истинна (равна 1). Сравнение ФПТ и 4И-НЕ реализаций БЗК представлено на рис. 3, 4.

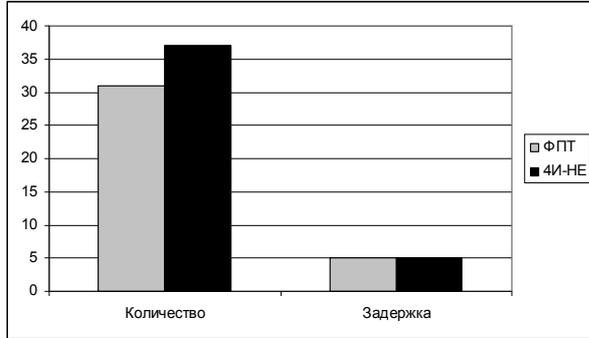


Рис. 3. Сравнение ФПТ/4И-НЕ реализаций восьмиразрядной конъюнкции при наличии парафазных входов

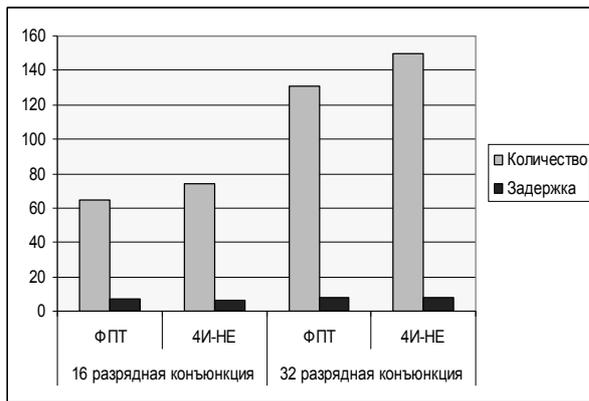


Рис. 4. Сравнение ФПТ/4И-НЕ реализаций 16 и 32-разрядной конъюнкции при наличии парафазных входов

4. Реализация блока функций БФ

Реализация одного блока функций БФ в ФПТ базисе имеет вид:

$$z = (z_1 z_{i01} \vee z_2 z_{i02})(z_3 z_{i03} \vee z_4 z_{i04}) \vee \dots \vee (z_{k-3} z_{i0(m-3)} \vee z_{k-2} z_{i0(m-2)}) \wedge (z_{k-1} z_{i0(m-1)} \vee z_k z_{i0m}). \quad (9)$$

Сравнение ФПТ/4И-НЕ реализаций восьмиразрядного блока функции при наличии парафазных входов представлено на рис.5:

Общий выигрыш в количестве элементов по сравнению с 4И-НЕ реализацией описывается выражением:

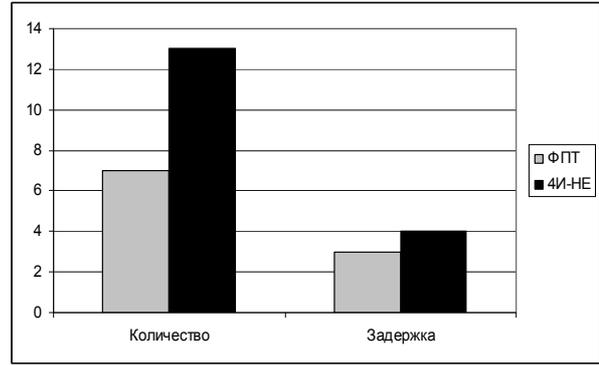


Рис. 5. Сравнение ФПТ/4И-НЕ реализаций восьмиразрядного блока функции БФ при наличии парафазных входов

$$\eta = \frac{n \cdot [(2^n - 1) + n]}{k \cdot (8n - 1)}. \quad (10)$$

Так, если, $n = m = 8, k = 10$ получаем:

$$\eta = \frac{8 \cdot [(2^8 - 1) + 8]}{10 \cdot (8 \cdot 8 - 1)} = \frac{2104}{630} = 3,34. \quad (11)$$

5. Парирование отказов

Оказывается, ФПТ элемент [6, 7] обладает свойством, которое раньше не входило в поле зрения исследователей: в случае любого однократного константного отказа (stuck-at fault) за счёт других трёх, не отказавших входов, его можно привести в любое из двух состояний – «0» или «1». Таким свойством не обладают существующие базисы И-НЕ, ИЛИ-НЕ.

Такое свойство сохраняется и для отказов типа замыканий (Bridging Faults) двух входов [11], причём тогда в некоторых случаях можно использовать все входы.

Рассмотрим парирование отказов в блоке конъюнкций. Для парирования одного отказа одного из входов элементов одного разряда блока вычислений конъюнкций (рис. 2) разрядов необходимо навязывать в любом случае ноль на выходе (чтобы не «мешать» вычислению остальных разрядов):

1. Допустим, имеется отказ типа «константа нуля» по входу x_i :

$$z_i = 0 \bar{x}_i \bar{x}_{id} \vee 0 \vee x_i \bar{x}_{id}. \quad (12)$$

Тогда, если $x_{id} = 0$

$$z_i = 1 \cdot 0 \vee 1 \vee x_i \cdot 0 = 0. \quad (13)$$

2. Допустим, имеется отказ типа «константа единицы» по входу x_i :

$$z_i = 1 \bar{x}_i \bar{x}_{id} \vee 1 \vee x_i \bar{x}_{id}. \quad (14)$$

Тогда, если $x_{io} = 0$:

$$z_i = 1 \cdot 1 \bar{x}_{id} \vee 1 \vee 0 \bar{x}_{id} = 0. \quad (15)$$

3. Допустим, имеется отказ типа «константа нуля» по входу x_{i0} :

$$z_i = \overline{x_i} \overline{0} \overline{x_{id}} \vee \overline{x_i} \vee \overline{0} \overline{x_{id}}. \quad (16)$$

Тогда, если $x_{id}=0$:

$$z_i = \overline{x_i} \overline{0} \overline{0} \vee \overline{x_i} \vee \overline{0} \overline{0} = 0. \quad (17)$$

4. Допустим, имеется отказ типа «константа единицы» по входу x_{i0} :

$$z_i = \overline{x_i} \overline{1} \overline{x_{id}} \vee \overline{x_i} \vee \overline{1} \overline{x_{id}}. \quad (18)$$

Тогда, если $x_{id}=0$, $x_i=0$:

$$z_i = \overline{0} \overline{1} \overline{0} \vee \overline{0} \vee \overline{1} \overline{0}; \quad (19)$$

$$z_i = 0 \vee \overline{1} \vee \overline{0} \overline{1} = 0. \quad (20)$$

5. Допустим, имеется отказ типа «константа нуля» по входу x_{i0} :

$$z_i = \overline{x_i} \overline{0} \overline{x_{id}} \vee \overline{x_i} \vee \overline{0} \overline{x_{id}}. \quad (21)$$

Тогда, если $x_{id}=0$:

Получаем

$$z_i = \overline{x_i} \overline{0} \overline{0} \vee \overline{x_i} \vee \overline{1} \overline{0}. \quad (22)$$

Тогда:

$$z_i = \overline{x_i} \overline{0} \overline{0} \vee \overline{x_i} \vee \overline{1} \overline{0} = 0. \quad (23)$$

Но в случаях 1, 2 имеется отказ по входу переменной x_i . А это общие входные переменные (для всех разрядов). Поэтому вероятность того, что отказ x_i , возникший на входе одного из блоков (рис. 2) будет распространён на все разряды будет весьма велика. Конечно, возможно в ряде случаев, что имеются и другие элементы, транслирующие x_i , конкретно на данный разряд (и отказ именно их!), но, если их нет – надо уменьшать разрядность всех блоков конъюнкций. В конце концов, это должна выяснять соответствующая диагностика.

Далее, единственный случай 4 требует $x_i = 0$ (остальные парируются настроечными сигналами только данной конъюнкции x_{i0} , x_{id}), а это общая переменная, то есть, разрядность всех блоков также уменьшится на 1, что не есть хорошо – из-за одного блока конъюнкций – снижать разрядность всех.

Тогда лучше отключить весь блок конъюнкций, в котором отказал данный разряд.

Парирование отказов вычисления конъюнкций – БК

Для этого дефектная конъюнкция, соответствующая, например, z_1 в выражении

$$Z_{Bj} = \overline{z_1} z_{101} \vee \overline{z_2} z_{102} \vee \dots \vee \overline{z_k} z_{10k} \quad (24)$$

отключается путём задания, например, $z_{101} = 0$, при этом число реализуемых конъюнкций уменьшается на одну.

В крайнем случае, когда остаются технические средства для всего одной работоспособной конъюнкции, устройство может быть использовано для микропрограммно-аппаратного вычисления за не-

сколько тактов, используя внешние технические средства, путём чередования циклов программирования каждой очередной конъюнкции и вычислений соответствующих ей функций.

Парирование отказов блоков функций - БФ

Может быть реализовано путём исключения этой функции, при этом число выходных функций уменьшается.

В крайнем случае, когда остаются технические средства для всего одной работоспособной функции, устройство может быть использовано для микропрограммно-аппаратного вычисления за несколько тактов, используя внешние технические средства, путём чередования циклов программирования конъюнкций и вычисления соответствующей им функции. То есть, за один цикл вычисляться будет только одна функция, возможно, зависящая от многих конъюнкций.

Возможна также работа на одной из двух оставшихся работоспособных доль БФ, путём отключения отказавшей (не формирующей необходимую единицу) сигналами $z_{i0i} = 0$, z_i .

Заключение

Предложен ДНФ – КЛБ на основе ФПТ элементов, реализующий систему логических функций в ДНФ, что существенно снижает аппаратные затраты на реализацию логики ПЛИС при не ухудшении быстродействия. Таких КЛБ в ПЛИС может быть десятки и сотни тысяч. Рекомендуются использование ДНФ-КЛБ для реализации отказоустойчивых ПЛИС с микропрограммно-аппаратным восстановлением логики. Целесообразны дальнейшие исследования по вопросам диагностики разработанных КЛБ и реализации «саморемонта» с возможным частичным структурным резервированием.

Литература

1. Look up table implementation of fast carry for adders and counters: US 005274581A, 28.12.1993.
2. Алексенко, А.Г. Проектирование радиоэлектронной аппаратуры на микропроцессорах [Текст] / А.Г. Алексенко, А.А. Голицын, А.Д. Иванов. – М.: Радио и связь, 1984. – 272 с.
3. Лазарев, В.Г. Построение программируемых управляющих устройств [Текст] / В.Г. Лазарев, Е.И. Пийль, Е.Н. Турута. – М.: Энергоиздат, 1984. – 193 с.
4. Программно-аппаратная реализация логических алгоритмов в микропроцессорных системах [Текст] / С.Ф. Тюрин, В.С. Харченко, Г.Н. Тимонькин, В.А. Мельников // Зарубежная радиоэлектроника. – 1992. – № 2. – С. 24–36.

5. Тюрин, С.Ф. Методы аппаратной поддержки логических алгоритмов в микропроцессорных системах [Текст] / С.Ф. Тюрин, Г.Н. Тимонькин, В.С. Харченко // Управляющие системы и машины. – 1993. – № 1. – С. 55 – 63.

6. Тюрин, С.Ф. Базисный элемент программируемых логических интегральных схем [Текст] / С.Ф. Тюрин, О.А. Громов // Вестник Ижевского государственного технического университета. – 2010. – № 3. – С. 122 – 126.

7. Пат. 2438234 Российская Федерация, Функционально-полный толерантный элемент [Текст] / №2010123392; заявл. 08.06.2010; опубл. 27.12.2011, Бюл. №36.

8. Программируемое логическое устройство / С.Ф. Тюрин, А.В. Набатов, О.А. Громов, А.В. Гре-

ков, Д.А. Карлов. – Заявка на выдачу патента РФ. 2012 г.

9. А.с. 1444892 СССР, G11C17/00, G06F7/00. Программируемое логическое устройство [Текст] / С.Ф. Тюрин, В.С. Харченко, С.Н. Ткаченко, В.Я. Жихарев, В.П. Улитенко. – Опубл. 1998; Бюл. № 46.

10. Пат. 2146840 Российская Федерация, Программируемое логическое устройство [Текст] / С.Ф. Тюрин, В.А. Несмелов, В.А. Харитонов и другие опубл.2000; БИ № 8.

11. Тюрин, С.Ф. Определение функционально-полных толерантных булевых функций четырех аргументов с учетом модели замыканий переменных [Текст] / С.Ф. Тюрин, А.В. Греков, О.А. Громов // Доклады Академии военных наук. – 2011. – № 5 (49). – С. 35–44.

Поступила в редакцию 20.01.2012

Рецензент: д-р техн. наук, проф. Ю.П. Кондратенко, Черноморский государственный университет им. Петра Могилы, Николаев, Украина.

ДНФ-КОНФІГУРОВАНІЙ ЛОГІЧНИЙ БЛОК ПЛІС З ВИКОРИСТАННЯМ ФУНКЦІОНАЛЬНО - ПОВНИХ ТОЛЕРАНТНИХ ЕЛЕМЕНТІВ

С.Ф. Тюрін, А.В. Набатов

У статті пропонується і аналізується конфігурований логічний блок(КЛБ), який реалізує логічні функції в ДНФ (диз'юнктивній нормальній формі) – ДНФ КЛБ. Основа блоку – апаратно реалізований алгоритм програмованої логічної матриці(АПЛА), причому в якості базисного елемента використовується функціонально-повний толерантний (ФПТ)елемент, що дозволяє одержати істотний вигравш у кількості елементів у порівнянні з еквівалентним по числу транзисторів базисом 4І-НІ і тим більше, 2І-НІ. Запропоновано процедури відбиття відмов з можливим переходом до мікропрограмно-апаратної реалізації логіки ПЛІС.

Ключові слова: програмована логічна інтегральна схема (ПЛІС), конфігурований логічний блок (КЛБ), досконала диз'юнктивна нормальна форма (СДНФ), диз'юнктивна нормальна форма (ДНФ), відмовостійкість, функціонально-повний толерантний (ФПТ) елемент, апаратно реалізований алгоритм програмованої логічної матриці (АПЛА).

DNF-CONFIGURABLE LOGIC BLOCK FPGA USING A FUNCTIONAL – FULL CELL TOLERANCE

S.F. Tyurin, A.V. Nabatov

The paper presents and analyzes the configurable logic block (KLB), which implements the logic functions in DNF (disjunctive normal form) - DNF KLB. It is based on hardware-implemented algorithm for programmable logic array (APLA), and functionally complete tolerance (FPT)element is used as the reference element, that provides a significant gain in the number of elements compared to an equivalent number of transistors on the basis of the 4AND -NOT and the more, 2AND -NOT. The procedures of failure parry with possible transition to the firm-ware - hardware implementation of logic FPGA are proposed.

Keywords: programmable logic integrated circuit (FPGA) configurable logic block (CLB), the perfect disjunctive normal form (PDFN), disjunctive normal form (DNF), functionally-complete fault-tolerable (FFT) element, implemented in hardware algorithm programmable logic array (APLA).

Тюрин Сергей Феофентович – д-р техн. наук, проф., заслуженный изобретатель РФ, проф. кафедры автоматизации и телемеханики Пермского национального исследовательского политехнического университета, Пермь, Россия.

Набатов Артем Владимирович – старший преподаватель кафедры автоматизированного управления войсками Пермского военного института внутренних войск МВД России, Пермь, Россия.