

ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

DESIGNING AND DIAGNOSTICS OF COMPUTER SYSTEMS

Научная статья

УДК 004.052.32+681.518.5

doi: 10.17223/19988605/58/10

Структуры схем коррекции ошибок на основе логического дополнения с контролем вычислений по коду с суммированием взвешенных переходов от разряда к разряду

Дмитрий Викторович Ефанов

Российский университет транспорта, Москва, Россия,

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия,

TrES-4b@yandex.ru

Аннотация. Предложены новые структуры отказоустойчивых цифровых вычислительных устройств и систем, в основе которых лежит использование принципа логического дополнения для фиксации искаженных сигналов и схем встроенного контроля. Последние реализуются с применением кода с суммированием взвешенных переходов от разряда к разряду в информационном векторе, при построении которого использована последовательность весовых коэффициентов, образующая ряд возрастающих степеней числа 2. Использование данного кода с суммированием позволяет обнаруживать любые комбинации искажений на выходах объекта диагностирования, за исключением одновременного искажения всех выходов, что на практике бывает достаточно редко. Дано описание четырех структур: структуры с двойной модульной избыточностью и контролем вычислений основным блоком по выбранному коду, структуры с двойной модульной избыточностью и контролем вычислений резервным блоком по выбранному коду, структуры с контролем вычислений основным блоком по выбранному коду и блоком фиксации искаженных сигналов на основе логического дополнения, структуры с блоком фиксации искаженных сигналов на основе логического дополнения с непосредственным контролем вычислений данным блоком. Приводятся примеры синтеза отказоустойчивых устройств и дана оценка их эффективности по сравнению с использованием традиционной структуры отказоустойчивых устройств и систем, основанной на тройной модульной избыточности с мажоритарной коррекцией сигналов. Освещены результаты экспериментов с контрольными комбинационными схемами LG'93 и MCNC Benchmarks, также показавшие эффективность предлагаемых отказоустойчивых структур. Использование принципа логического дополнения позволяет синтезировать отказоустойчивые цифровые устройства и системы, в которых не требуется прямого резервирования и внесения модульной избыточности, что на практике может давать существенное снижение структурной избыточности конечного объекта.

Ключевые слова: отказоустойчивые цифровые устройства и системы; модульная избыточность; схема встроенного контроля; коррекция сигналов; контроль вычислений; код с суммированием взвешенных переходов; структурная избыточность

Для цитирования: Ефанов Д.В. Структуры схем коррекции ошибок на основе логического дополнения с контролем вычислений по коду с суммированием взвешенных переходов от разряда к разряду // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2022. № 58. С. 105–125. doi: 10.17223/19988605/58/10

Error correction circuits structures based on Boolean complement with calculation checking by code with summation of weighted transitions from bit to bit

Dmitry V. Efanov

*Russian University of Transport, Moscow, Russian Federation,
Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation,
TrES-4b@yandex.ru*

Abstract. Methods for introducing modular redundancy are widely used in fault-tolerant digital devices and systems synthesis. Such methods involve the use of the original device's exact copies (analogues) and error correction circuits. There are structures based on the introduction of triple and double modular redundancy widespread and knowing. Research shows that fault-tolerant structures can be obtained without using modular redundancy techniques. At the same time, it is possible to achieve a decrease in the introduced hardware redundancy to endow the device with the property of fault tolerance concerning methods based on the introduction of modular redundancy.

An effective technical solution is the use of the logical complement principle with the built-in control circuits organization, the diagnostics object, or a Boolean complement block in the fixing circuit of distorted signals. When organizing built-in control circuits using the Boolean complement method, it is possible to consider the structural features of the diagnosed object itself, and this makes it possible to reduce the structural redundancy relative to known methods.

The article proposes standard structures of fault-tolerant devices based on the weight-transition sum code from bit to bit. Such a code is obtained by dividing the bits of the data vector into $k = m - 1$ pairs, starting with the least significant bit (pairs (f_1, f_2) , (f_2, f_3) , ..., (f_{m-1}, f_m)), are formed), assigning the pairs of weight coefficients from a series of increasing powers of the number 2 ($2^0, 2^1, \dots, 2^{m-3}, 2^{m-2}$), the subsequent addition of the weighting coefficients values of the digits pairs according to the formula $W = \sum_{i=1}^{m-1} w_{i,i+1} q_{i,i+1}$, where $q_{i,i+1} = f_i \oplus f_{i+1}$ is the activating function the transition between the digits f_i and f_{i+1} . This code is designated as a T_m -code. It has $k = m - 1$ check bit, each of which is implemented as a convolution modulo $M = 2$ bits in the corresponding pair. In previous author research, it was proved that this code detects any errors in data vectors, except for errors with maximum multiplicity $d = m$. The noted property can be effectively used in the built-in control circuits synthesis, if exclude the diagnostic object faults influence immediately all its outputs, any combination of distortions can be detected. The use of the T_m -code in the built-in control circuits synthesis gives a decrease in the introduced structural redundancy with the duplication method use. This feature is proposed to be used in the synthesis of fault-tolerant devices.

The paper proposes four typical fault-tolerant structures based on the use of the T_m -code. The 1st two structures are based on the use of the duplication with computations control principle by one of the devices. In the first case, the calculations at the outputs of the main block are controlled. In the second, the calculations at the outputs of the additional block are controlled. The differences between the structures are in the method of signal correction when fixing computational errors. The other two structures are based on the use of only one source device and the implementation of a special block for fixing distorted signals using the Boolean complement method. In the first case, the calculations by the main unit are controlled by the T_m -code, and the fixation of distortions is performed in fixing the distorted signals. In the second case, the calculations are controlled in the block for fixing the distorted signals. The use of Boolean complement makes it possible to synthesize many variants of blocks for fixing distorted signals, which makes it possible to choose the variant that gives the least introduced structural redundancy.

The block for fixing the distorted bits includes a Boolean complement block and a block for calculating correction functions. In the second case, this block also includes a built-in T_m -code control circuit, which controls the calculations at the control logic block outputs. The task of synthesizing a block for fixing distorted bits is to synthesize a built-in control circuit, a block for calculating correction functions, and a Boolean complement block. The first two, as shown in the article, have a standard implementation. For the Boolean complement block synthesis, the article proposes two algorithms. An example of the synthesis of all four structures is given.

The theoretical results are complemented by the results obtained in experiments with combinational benchmarks LG'93 and MCNC Benchmarks. The results show the effectiveness of the proposed structures in comparison with the known ones.

The use of Boolean complement and code methods for the self-checking and fault-tolerant digital devices and systems synthesis makes it possible to expand the number of methods for their implementation and to minimize the structural redundancy indicators. The results obtained in the article should be taken into account in practice when choosing an approach to the synthesis of fault-tolerant digital devices and systems.

Keywords: fault-tolerant digital devices and systems; modular redundancy; built-in control circuit; correction of signals; checking of calculations; code with summation of weighted transitions; structural redundancy

For citation: Efanov, D.V. (2022) Error correction circuits structures based on Boolean complement with calculation checking by code with summation of weighted transitions from bit to bit. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika – Tomsk State University Journal of Control and Computer Science*. 58. pp. 105–125. doi: 10.17223/19988605/58/10

Методы коррекции ошибок в вычислениях повсеместно используются при построении надежных цифровых устройств и систем [1–4]. Все они подразумевают внесение избыточности по определенным правилам, например внесение модульной избыточности с проверкой вычислений и выбором верного значения по принципу большинства (мажоритарный принцип) или использование предварительно при синтезе устройства помехоустойчивого кодирования состояний или выходов устройства и т.д. [5–14].

Как показано в ряде работ [15, 16], для синтеза устройств и систем, не чувствительных к проявлениям ошибок, возможно применение принципа логического дополнения с организацией схем встроенного контроля (СВК) либо объекта диагностирования, либо блока логического дополнения в схеме фиксации искаженных сигналов (ФИС). Это позволяет учитывать особенности структуры самого объекта диагностирования и снижать структурную избыточность, необходимую для реализации отказоустойчивой системы.

Метод логического дополнения описан в [17, 18] и подразумевает преобразование значений функций некоторого объекта диагностирования (исходного устройства) в СВК в функции специального вида. Такое преобразование осуществляется с использованием двухвыходовых элементов сложения по модулю $M = 2$ (элементов XOR) и блока логического дополнения, имеющего такое же (или меньшее) число выходов, как и исходное устройство. Преобразования, таким образом, осуществляются по формуле: $f_i \oplus g_i$, $i \in \{1, 2, \dots, m\}$, m – число выходов исходного устройства. В качестве диагностического признака может быть выбран признак принадлежности вектора функций, формируемых на выходах каскада элементов преобразования, заранее выбранному избыточному блочному коду [19], принадлежность преобразованной функции к классу самодвойственных функций алгебры логики [20], контроль по обоим этим признакам [21] и т.д. Интересным является использование в качестве диагностического признака принадлежности вектора, формируемого на выходах каскада элементов преобразования, коду с суммированием взвешенных переходов от разряда к разряду в информационном векторе, при построении которого используется последовательность возрастающих степеней числа 2: $\{2^0, 2^1, \dots, 2^{m-3}, 2^{m-2}\}$ [22, 23]. Такой код имеет избыточность, практически сравнимую с кодом с повторением: $k = m - 1$. При этом данным кодом, обозначим его как T_m -код, обнаруживаются любые сочетания искажений информационных разрядов, кроме одновременного искажения всех разрядов информационного вектора. Как показано в [22, 23], использование T_m -кода при синтезе СВК позволяет получать самопроверяемые логические устройства автоматики и вычислительной техники со структурной избыточностью, меньшей, чем при использовании принципа дублирования со сравнением результатов вычислений. В [24, 25] предложено применять T_m -код совместно с внесением двойной модульной избыточности при синтезе отказоустойчивых цифровых устройств и систем, а также приведены результаты экспериментов, позволяющие судить об эффективности применения такого подхода. Например, для представленных 25 комбинационных устройств было получено среднее значение уменьшения структурной избыточности примерно до 94,614% по сравнению с применением традиционного подхода реализации мажоритарной структуры (структуры с тройной модульной избыточностью). Для отдельных устройств были получены показатели уменьшения структурной избыточности более чем на 10%.

В данной статье рассматриваются возможности использования T_m -кода при реализации отказоустойчивых устройств по принципу логического дополнения с контролем вычислений исходным устройством либо блоком логического дополнения в схеме ФИС. Применение логического дополнения позволяет уменьшать сложность технической реализации отказоустойчивых устройств и систем

по сравнению с подходами, подразумевающими внесение модульной избыточности, при высокой корректирующей способности.

1. Отказоустойчивая структура на основе дублирования с контролем вычислений по коду с суммированием взвешенных переходов

Исследования [26, 27] показывают, что при синтезе СВК могут эффективно использоваться разнообразные равномерные избыточные коды, обладающие различными избыточностью и характеристиками обнаружения ошибок. Среди таких кодов можно выделить специальный код с суммированием взвешенных переходов от разряда к разряду в информационном векторе, который строится по следующим правилам [22, 23]:

1. Переходам от разряда к разряду информационного вектора присваиваются весовые коэффициенты $w_{i,i+1}$ из ряда возрастающих степеней числа 2: $[w_{i,i+1}] = [w_{1,2}, w_{2,3}, \dots, w_{m-2,m-1}, w_{m-1,m}] = [2^{m-2}, 2^{m-3}, \dots, 2^1, 2^0]$. Будем их приписывать переходам в информационном векторе, начиная справа налево.

2. Подсчитывается суммарный вес активных переходов:

$$W = \sum_{i=1}^{m-1} w_{i,i+1} q_{i,i+1}, \quad (1)$$

где $q_{i,i+1} = f_i \oplus f_{i+1}$ – функция активации перехода между разрядами f_i и f_{i+1} .

3. Полученное число представляется в двоичном виде и записывается в разряды контрольного вектора.

К примеру, получим контрольный вектор для информационного вектора $\langle f_1 f_2 f_3 f_4 f_5 f_6 \rangle = \langle 101101 \rangle$ T_6 -кода: $W = 2^4 \cdot 1 + 2^3 \cdot 1 + 2^2 \cdot 0 + 2^1 \cdot 1 + 2^0 \cdot 1 = 16 + 8 + 2 + 1 = 27$, $[W]_2 = \langle 11011 \rangle$.

Как отмечалось ранее, T_m -код имеет $k = m - 1$ контрольный разряд. Значения контрольных разрядов могут быть определены по формулам

$$\begin{aligned} h_1 &= f_1 \oplus f_2; \\ h_2 &= f_2 \oplus f_3; \\ &\dots \\ h_{m-1} &= f_{m-1} \oplus f_m. \end{aligned} \quad (2)$$

Так как для получения значений контрольных разрядов кода с суммированием взвешенных переходов используются только операции сложения по модулю $M = 2$, структура кодера данного кода будет стандартной и содержать в себе $m - 1$ элемент сложения по модулю два. Наличие стандартной структуры кодера позволяет синтезировать типовую структуру СВК по данному коду, подробно описанную в [22, 23].

T_m -код будет обнаруживать любые искажения в контролируемом кодовом векторе, за исключением ошибок с максимальной кратностью $d = m$. Это объясняется тем, что значение суммарного веса информационного вектора, вычисленного по формуле (1), не изменится только в том случае, если оно будет вычислено для двух векторов с полностью противоположными значениями разрядов. Такая особенность T_m -кода позволяет весьма эффективно использовать его для организации СВК логических устройств. При этом на структуры контролируемых устройств накладывается только одно ограничение – отсутствие путей от каких-либо внутренних логических элементов, ведущих сразу же ко всем их выходам (можно сказать, что это структурное ограничение). Однако даже при наличии таких элементов может быть проверено условие невозможности одновременного искажения всех m выходов устройства:

$$\frac{\partial f_1}{\partial y_\alpha} \cdot \frac{\partial f_2}{\partial y_\alpha} \cdot \dots \cdot \frac{\partial f_m}{\partial y_\alpha} = 0, \quad (3)$$

где y_α – функция, реализуемая на выходе логического элемента G_α , связанного путями со всеми выходами устройства.

Если выражение в левой части формулы (3) равно нулю, то не существует ни одного входного набора, на котором происходит трансляция ошибок на все выходы устройства.

На рис. 1, 2 изображены отказоустойчивые структуры, основанные на использовании принципов дублирования (внесении двойной модульной избыточности) с контролем вычислений одним из устройств (исходным $F(x)$ или его копией $F^*(x)$).

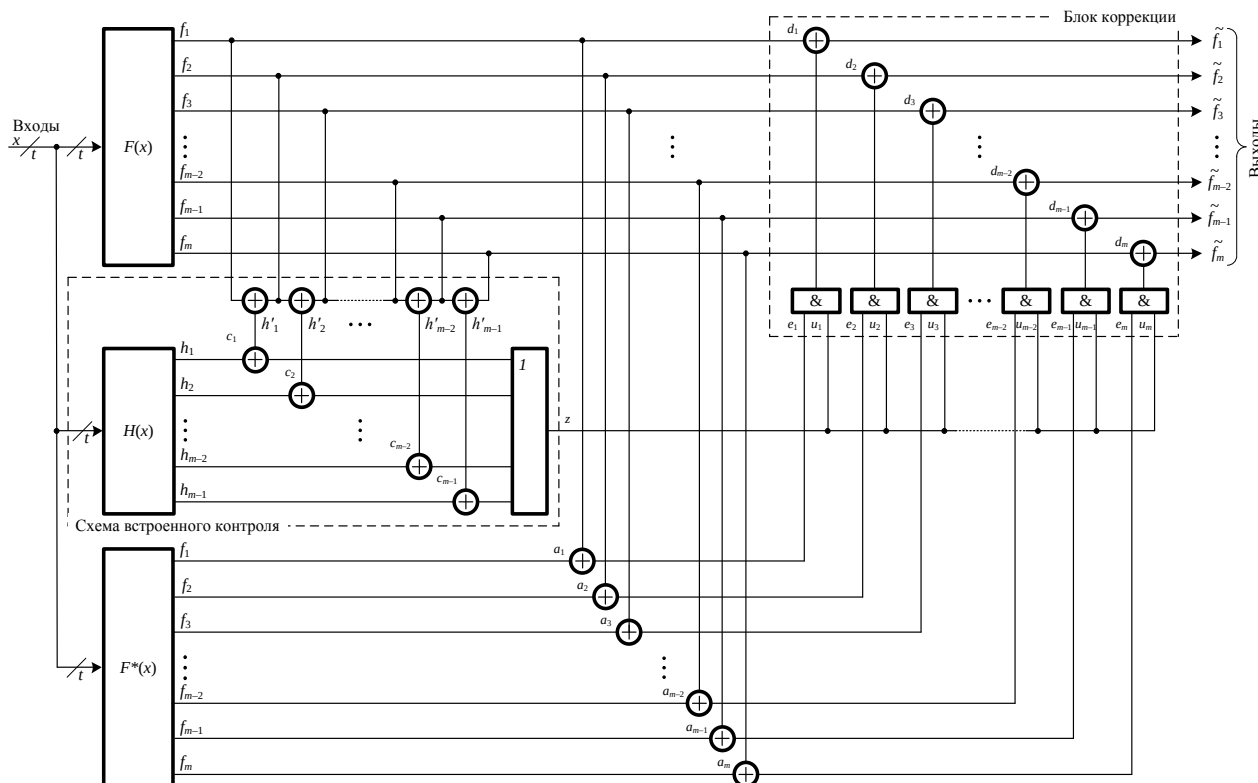


Fig. 1. Fault-tolerant structure based on duplication with checking of calculations by the main device according to the T_m -code

В обеих структурах для выявления ошибок на выходах исходного объекта $F(x)$ используются дополнительное устройство $F^*(x)$, являющееся копией исходного устройства, и каскад элементов сложения по модулю $M = 2$ $a_1, a_2, \dots, a_{m-1}, a_m$ для выявления выходов с ошибками. Если на каком-либо из выходов присутствует ошибка, то на соответствующем выходе элемента сравнения будет присутствовать единичное значение. Такая организация позволяет выявлять любые одиночные неисправности на выходах объекта $F(x)$. Для определения устройства, на выходах которого возникла ошибка, используется СВК по T_m -коду. В первом случае осуществляется контроль вычислений исходным объектом $F(x)$, а во втором – объектом $F^*(x)$. Наличие СВК позволяет установить, на выходах какого из объектов возникла ошибка, и осуществить коррекцию только в том случае, если ошибка возникла на выходах основного объекта $F(x)$.

СВК содержит каскад сумматоров по модулю $M = 2$, позволяющих реализовать кодер T_m -кода с выходами $h'_1, h'_2, \dots, h'_{m-2}, h'_{m-1}$, блок $H(x)$, вычисляющий аналогичные функции $h_1, h_2, \dots, h_{m-2}, h_{m-1}$, но по значениям входных воздействий, а также схему сравнения, образованную каскадом сумматоров $c_1, c_2, \dots, c_{m-2}, c_{m-1}$ и элементом логического сложения с $m - 1$ входами (элементом OR), на выходах которого формируется сигнал z , свидетельствующий об отсутствии или наличии ошибки. Схема сравнения может быть реализована в виде полностью самопроверяемого устройства с использованием модулей сжатия парафазных сигналов (two-rail checkers [28]).

Для коррекции ошибок используется специализированный блок коррекции сигналов, включающий в себя элементы принятия решения о коррекции (элементы *AND*) и элементы коррекции сигналов (элементы *XOR* $d_1, d_2, \dots, d_{m-1}, d_m$). Элементы принятия решения о коррекции имеют по два входа:

на первые входы поступают сигналы $e_1, e_2, \dots, e_{m-1}, e_m$ от элементов сравнения $a_1, a_2, \dots, a_{m-1}, a_m$ и определяют те выходы, на которых необходимо произвести коррекцию сигналов. На вторые входы поступают сигналы $u_1, u_2, \dots, u_{m-1}, u_m$ от СВК: $u_1 = u_2 = \dots = u_{m-1} = u_m = z$. Коррекция в случае структуры рис. 1, когда СВК контролирует вычисления основным комплектом, осуществляется в том случае, если сигнал $z = 1$. Поэтому в ней на входах каждого элемента AND рассчитывается функция $ze_i, i = \overline{1, m}$. Коррекция в случае структуры рис. 2, когда вычисления контролируются резервным комплектом, осуществляется только в том случае, если сигнал $z = 0$ (коррекция производится только при наличии ошибок в вычислениях основным блоком). Поэтому в данной структуре на входах каждого элемента AND рассчитывается функция $\bar{z}e_i, i = \overline{1, m}$.

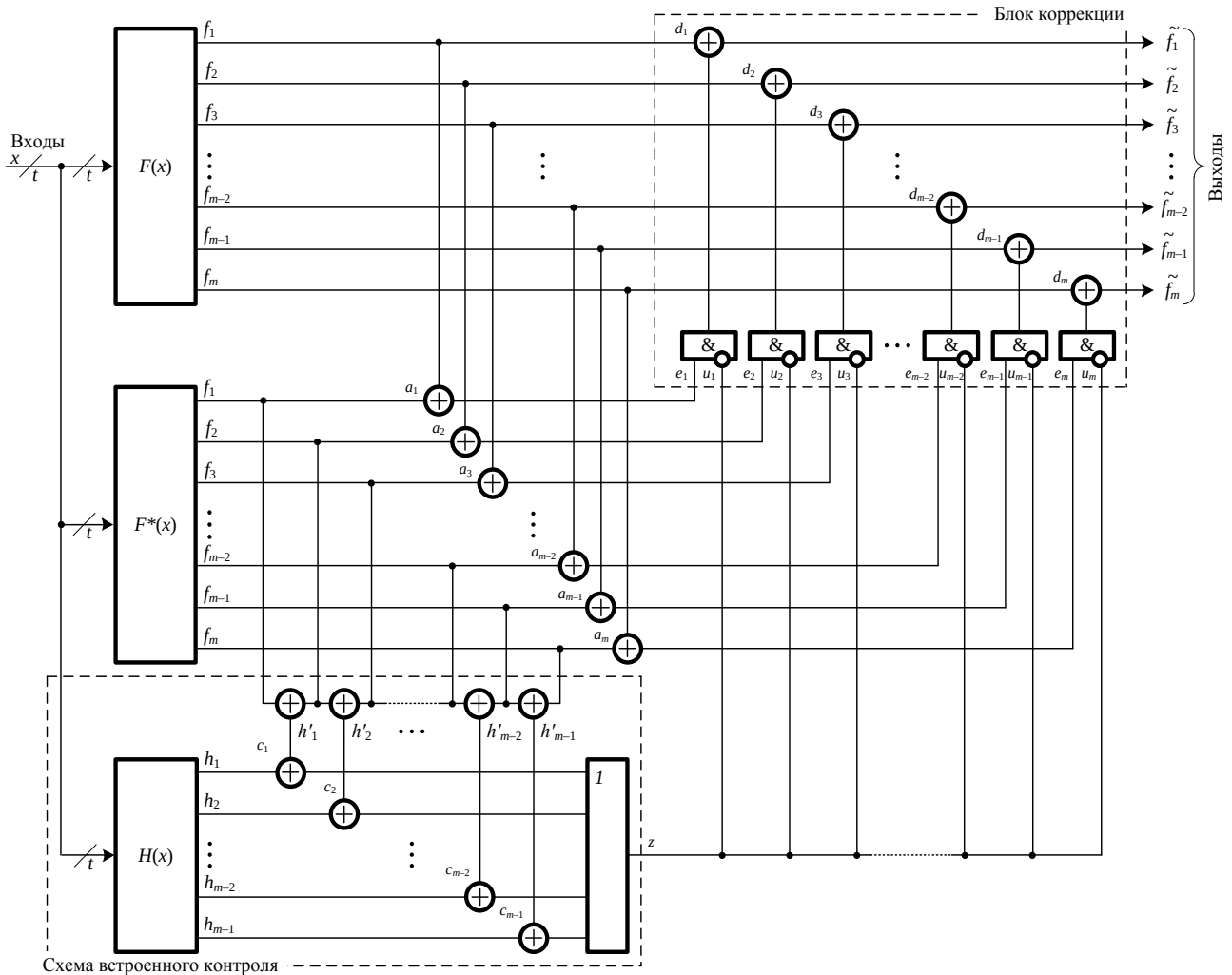


Рис. 2. Отказоустойчивая структура на основе дублирования с контролем вычислений
дополнительным устройством по T_m -коду

Fig. 2. Fault-tolerant structure based on duplication with checking of calculations by an additional device according to the T_m -code

Следует отметить, что структуры, приведенные на рис. 1, 2, не защищены от неисправностей элементов блока коррекции сигналов. Такой же недостаток присущ и традиционной схеме мажоритарной коррекции сигналов [1, 2, 6]. На практике данный недостаток нивелируется применением высоконадежных элементов при синтезе блока коррекции сигналов, имеющих низкие интенсивности потоков отказов, внутреннее резервирование и реализуемых в виде полностью самопроверяемых устройств [29].

Обе структуры имеют высокую корректирующую способность на выходах исходного объекта за счет применения T_m -кода.

2. Отказоустойчивая структура на основе логического дополнения с контролем вычислений по коду с суммированием взвешенных переходов

Исследования показывают, что могут быть реализованы отказоустойчивые устройства, имеющие в своем составе только один блок $F(x)$. При этом возможно уменьшение структурной избыточности конечного устройства не только по сравнению с применением мажоритарной структуры, но и по сравнению с описанными выше структурами, основанными на принципе дублирования. Новые структуры реализуются за счет применения логического дополнения при синтезе блока ФИС. Для контроля корректности вычислений используется СВК на основе T_m -кода. В структуре рис. 3 контролируются вычисления исходным объектом $F(x)$. В структуре рис. 4 контролируются вычисления блоком логического дополнения $G(x)$. Для этого используется блок $H^*(x)$.

Отличием обеих структур – рис. 3 и рис. 4 – от структур, приведенных на рис. 1, 2, является использование блока ФИС.

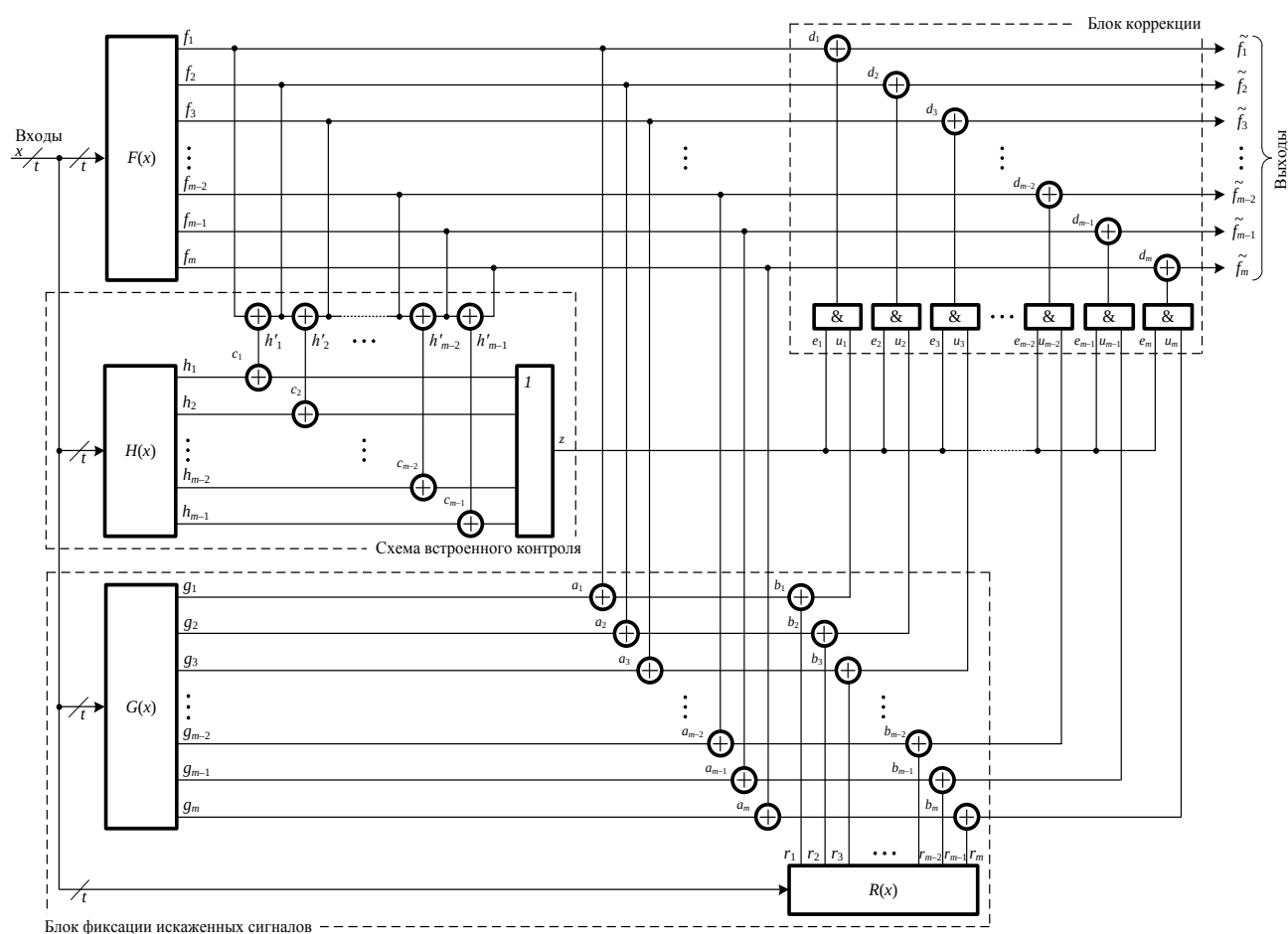


Рис. 3. Отказоустойчивая структура на основе логического дополнения с контролем вычислений исходным устройством по T_m -коду

Fig. 3. Fault-tolerant structure based on Boolean complement with checking of calculations by the source device according to the T_m -code

В структуру блока ФИС для структуры, приведенной на рис. 3, входит блок логического дополнения $G(x)$, на выходах которого вычисляются специальные функции дополнения $g_1, g_2, \dots, g_{m-1}, g_m$, а также блок вычисления функций коррекции $R(x)$. Значения функций $g_1, g_2, \dots, g_{m-1}, g_m$ сравниваются со значениями с одноименных выходов блока $F(x)$. Для вычисления функций активизации коррекции использован блок $R(x)$, вычисляющий функции $r_1, r_2, \dots, r_{m-1}, r_m$, а также каскад сумматоров по модулю $M = 2$ $b_1, b_2, \dots, b_{m-1}, b_m$. На выходах последних формируются значения функций ошибок:

$$\begin{aligned} e_1 &= f_1 \oplus g_1 \oplus r_1 = 0; \\ e_2 &= f_2 \oplus g_2 \oplus r_2 = 0; \\ &\dots \\ e_m &= f_m \oplus g_m \oplus r_m = 0. \end{aligned} \quad (4)$$

Если в результате неисправности в блоке $F(x)$ искажается значение функции f_i , то, как следует из (4), $e_i = 1$, и активизируются вход и выход соответствующего элемента логического умножения в блоке коррекции сигналов (при этом на выходе СВК блока $F(x)$ сигнал $z = 1$). Осуществляется исправление неверного значения функции f_i .

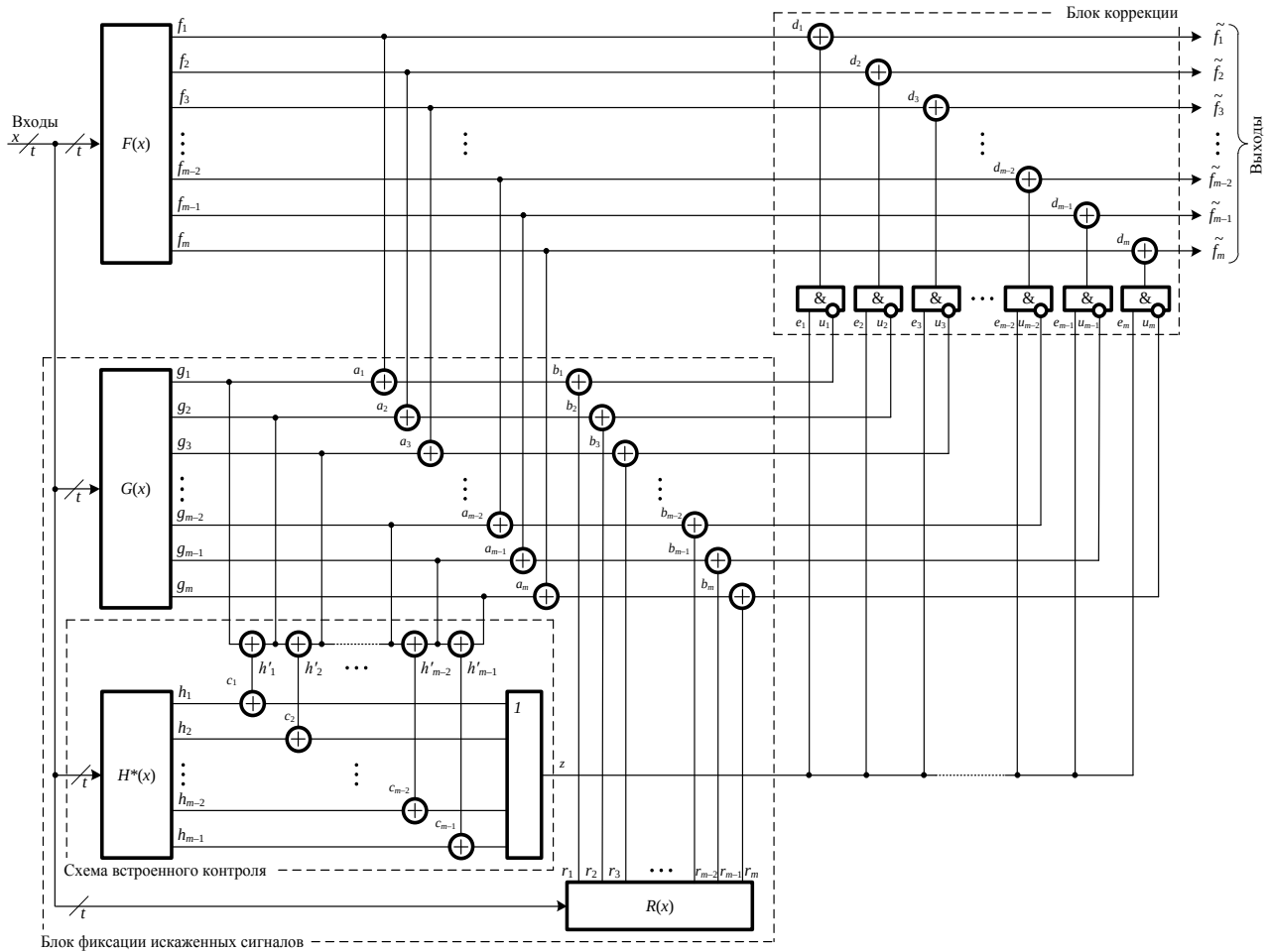


Рис. 4. Отказоустойчивая структура на основе логического дополнения с контролем вычислений блоком дополнения по T_m -коду

Fig. 4. Fault-tolerant structure based on Boolean complement with checking of computations by the complement block by T_m -code

В структуре рис. 4 вычисления основным устройством не контролируются, а блок ФИС, в отличие от структуры рис. 3, также содержит СВК для блока логического дополнения $G(x)$, реализуемую по T_m -коду (блок $H^*(x)$ формирует значения разрядов T_m -кода по значениям входов системы). СВК позволяет зафиксировать факт корректной или некорректной работы блока контрольной логики и сформировать соответствующий сигнал z .

3. Синтез блока фиксации искаженных сигналов и схемы встроенного контроля

Для структур рис. 1, 2 требуется реализация только СВК, в которой все элементы являются стандартными, кроме блока $H(x)$. Он синтезируется достаточно просто. На каждом входном воздей-

ствии рассчитываются значения функций $h_1, h_2, \dots, h_{m-1}$. Осуществляется минимизация полученных функций любым из известных методов [30].

Для структур рис. 3, 4 требуется синтез блоков дополнительной логики $G(x)$ и блока $H(x)$. Последний для структуры рис. 3 совпадает с блоком контрольной логики для структур рис. 1, 2. Для структуры рис. 4 он синтезируется аналогичным образом, за тем лишь исключением, что функции $h_1, h_2, \dots, h_{m-1}$ рассчитываются по выходам блока $G(x)$.

При синтезе устройства $G(x)$ в структурах рис. 3, 4 необходимо определить функции $g_1, g_2, \dots, g_{m-1}, g_m$ через функции $r_1, r_2, \dots, r_{m-1}, r_m$: $g_i = f_i \oplus r_i$. Каждая функция r_i может быть выбрана произвольным образом независимо от того, какими являются функции f_i и g_i . От выбранного варианта функции r_i зависит сложность реализации функции g_i . При этом никакой зависимости между функциями g_i и r_i с различными индексами i не существует. Однако наиболее простым способом вычисления функций r_i является придание им значений входных переменных. Это позволяет фактически не синтезировать блок $R(x)$, а реализовывать его в виде коммутатора входных сигналов.

Рассмотрим два алгоритма синтеза блока $G(x)$.

Алгоритм 1. Правила синтеза блока $G(x)$ при использовании в качестве функций $\{r_1, r_2, \dots, r_{m-1}, r_m\}$ одной из входных переменных:

1. Последовательно рассматриваются функции выходов блока $F(x)$ f_i ($i \in \{1, 2, \dots, m\}$).
2. Полагается $i = 1$.
3. Последовательно выбираются значения функции корректирующей переменной $r_j = x_j$ ($j \in \{1, 2, \dots, t\}$).
4. Полагается $j = 1$.
5. Вычисляются функции дополнения для каждого из выходов блока $F(x)$ f_i : $g_{i,j} = f_i \oplus r_j$ ($i \in \{1, 2, \dots, m\}$).
6. Функции $g_{i,j}$ оптимизируются (на данном этапе можно провести оптимизацию функций как системы [30]), что дает возможность синтеза блока $G(x)$ в выбранном элементном базисе.

Алгоритм 1 позволяет достаточно просто синтезировать блок дополнительной логики. При этом для каждого выхода устройства $G(x)$ может быть подобрана та переменная, которая дает наиболее эффективное с позиции сложности реализации техническое решение.

Алгоритм 2. Правила синтеза блока $G(x)$ при использовании в качестве функций $\{r_1, r_2, \dots, r_{m-1}, r_m\}$ входных переменных:

1. Последовательно рассматриваются функции выходов блока $F(x)$ f_i ($i \in \{1, 2, \dots, m\}$).
2. Полагается $i = 1$.
3. Последовательно выбираются значения функции корректирующей переменной $r_j = x_j$ ($j \in \{1, 2, \dots, t\}$).
4. Полагается $j = 1$.
5. Вычисляется функция дополнения $g_{i,j} = f_i \oplus r_j$.
6. Функция $g_{i,j}$ оптимизируется любыми известными методами [30].
7. Полагается $j := j + 1$.
8. Проверяется условие достижения конечной входной переменной $j > t$? Если нет, то возвращаемся к п. 5 данного алгоритма, в противном случае переходим к п. 9.
9. Полагается $i := i + 1$.
10. Проверяется условие достижения конечной функции выхода блока $F(x)$ $i > m$? Если нет, то возвращаемся к п. 3 данного алгоритма, в противном случае переходим к п. 11.
11. Для каждой функции $g_{i,j}$ выбираем ту входную переменную, коррекция по которой дает наименьшую сложность реализации в выбранном элементном базисе.
12. Синтезируется блок реализации системы функций дополнений $G(x)$.

Следует отметить, что в качестве функций $\{r_1, r_2, \dots, r_{m-1}, r_m\}$ могут быть выбраны произвольные функции, что увеличивает число вариантов синтеза отказоустойчивых цифровых устройств, при этом может быть выбран способ, дающий наименьшую сложность их реализации.

Приведем далее примеры синтеза отказоустойчивых устройств по всем четырем рассмотренным ранее структурам.

4. Пример синтеза отказоустойчивых структур

Рассмотрим в качестве примера процедуры синтеза структур, представленных на рис. 1–4, для комбинационного логического устройства, описываемого формулами

$$\begin{cases} f_1 = \overline{x_1} \overline{x_3} x_4 \vee \overline{x_1} \overline{x_2} \overline{x_4} \vee \overline{x_1} x_2 \overline{x_4} \vee \overline{x_1} x_2 x_3 \vee \overline{x_1} x_3 \overline{x_4}; \\ f_2 = \overline{x_1} x_2 x_3 \vee \overline{x_1} x_3 \overline{x_4} \vee \overline{x_1} x_2 \overline{x_4} \vee \overline{x_1} x_2 x_3 \vee \overline{x_1} x_2 x_3 \vee \overline{x_1} x_2 x_3 x_4; \\ f_3 = \overline{x_3} x_4 \vee \overline{x_1} x_2 x_4 \vee \overline{x_2} x_3 \overline{x_4} \vee \overline{x_1} x_2 x_4; \\ f_4 = \overline{x_1} x_2 x_3 \vee \overline{x_1} x_2 x_4 \vee \overline{x_2} x_3 \overline{x_4} \vee \overline{x_1} x_2 x_3 \vee \overline{x_1} x_2 x_4; \\ f_5 = \overline{x_1} x_2 x_4 \vee \overline{x_1} x_2 x_3 \vee \overline{x_1} x_2 x_3 \vee \overline{x_1} x_2 x_3 x_4. \end{cases} \quad (5)$$

Для синтеза структур, основанных на использовании двойной модульной избыточности (см. рис. 1, 2), требуется синтезировать только блок контрольной логики $H(x)$, а остальные компоненты являются типовыми. В табл. 1 приводится описание исходного блока $F(x)$ в виде таблицы истинности, а также контрольных функций, вычисляемых блоком $H(x)$. Непосредственно из табл. 1 путем оптимизации получаются логические выражения для функций h_1, h_2, h_3 и h_4 , по которым синтезируется блок $H(x)$.

Таблица 1

Описание комбинационного устройства и функций для контроля блока $F(x)$ или его копии $F^*(x)$ по T_m -коду

$x_4 x_3 x_2 x_1$	$f_5 f_4 f_3 f_2 f_1$	Активные переходы	Формула подсчета веса	W	$h_4 h_3 h_2 h_1$
0000	01101	$q_{1,2}, q_{2,3}, q_{4,5}$	$w_{1,2} + w_{2,3} + w_{4,5}$	11	1011
0001	01110	$q_{1,2}, q_{4,5}$	$w_{1,2} + w_{4,5}$	9	1001
0010	10111	$q_{3,4}, q_{4,5}$	$w_{3,4} + w_{4,5}$	12	1100
0011	00010	$q_{1,2}, q_{2,3}$	$w_{1,2} + w_{2,3}$	3	0011
0100	10110	$q_{1,2}, q_{3,4}, q_{4,5}$	$w_{1,2} + w_{3,4} + w_{4,5}$	13	1101
0101	00100	$q_{2,3}, q_{3,4}$	$w_{2,3} + w_{3,4}$	6	0110
0110	11001	$q_{1,2}, q_{3,4}$	$w_{1,2} + w_{3,4}$	5	0101
0111	11001	$q_{1,2}, q_{3,4}$	$w_{1,2} + w_{3,4}$	5	0101
1000	11000	$q_{3,4}$	$w_{3,4}$	4	0100
1001	10100	$q_{2,3}, q_{3,4}, q_{4,5}$	$w_{2,3} + w_{3,4} + w_{4,5}$	14	1110
1010	11010	$q_{1,2}, q_{2,3}, q_{3,4}$	$w_{1,2} + w_{2,3} + w_{3,4}$	7	0111
1011	01011	$q_{2,3}, q_{3,4}, q_{4,5}$	$w_{2,3} + w_{3,4} + w_{4,5}$	12	1100
1100	00111	$q_{3,4}$	$w_{3,4}$	4	0100
1101	11111	—	—	0	0000
1110	01000	$q_{3,4}, q_{4,5}$	$w_{3,4} + w_{4,5}$	12	1100
1111	00110	$q_{1,2}, q_{3,4}$	$w_{1,2} + w_{3,4}$	5	0101

Часто при синтезе цифровых устройств используются системы автоматизированного проектирования, в которые интегрированы средства оптимизации функций и оценки сложности их технической реализации [31, 32]. Воспользуемся известным интерпретатором SIS, синтезируем в нем исходные устройства $F(x)$, отказоустойчивую структуру на основе тройной модульной избыточности, а также предложенные в настоящей работе отказоустойчивые структуры с контролем вычислений по T_m -коду. При этом будем использовать библиотеку стандартных функциональных элементов stdcell2_2.genlib, а саму процедуру синтеза будем проводить по табличным формам задания устройств (в виде файлов *.pla). SIS позволяет не только синтезировать сами цифровые устройства, но и оценивать сложность их технической реализации в условных единицах площади, занимаемой

устройством на кристалле. Это дает возможность сравнения различных отказоустойчивых структур между собой. Далее не будем приводить сами устройства ввиду их громоздкости, а приведем только численные данные о показателях сложности реализации устройств.

Сложность исходного блока $F(x)$ равна $L_{F(x)} = 1080$. Мажоритарный элемент сравнения имеет сложность $L_{\geq 2} = 136$. Отсюда следует, что сложность отказоустойчивого устройства, реализованного по мажоритарной структуре, равна

$$L_{TMR} = 3L_{F(x)} + 5L_{\geq 2} = 3 \cdot 1080 + 5 \cdot 136 = 3920.$$

Оценим сложности отказоустойчивых устройств, реализованных по структурам рис. 1 и рис. 2. При этом учтем, что сложности стандартных функциональных элементов равны: элементов сложения по модулю два – $L_{XOR} = 40$, четырехвходового элемента ИЛИ – $L_{4OR} = 48$, сложность элемента логического умножения (как с инверсией на одном входе, так и без нее) – $L_{AND} = 32$. Оптимизация функций блока $H(x)$ с применением стандартной процедуры simplify дает следующий результат: $L_{H(x)} = 560$. С учетом полученного рассчитаем сложность структур, синтезируемых на основе двойной модульной избыточности с контролем вычислений одним из блоков по T_m -коду [25]:

$$L_{DMR} = 2L_{F(x)} + L_{H(x)} + 18L_{XOR} + 5L_{AND} + L_{4OR} = 2 \cdot 1080 + 560 + 18 \cdot 40 + 5 \cdot 32 + 48 = 3648.$$

Площади структур устройств, полученные на основе дублирования с контролем вычислений одним из блоков, равны и составляют 96,061% от площади устройства, реализованного в соответствии с мажоритарной структурой.

Синтезируем отказоустойчивое устройство по структуре, представленной на рис. 3. В ней схема блока $H(x)$ остается той же. Требуется синтезировать блоки $G(x)$ и $R(x)$ в структуре блока ФИС. Как отмечалось выше, фактически блок $R(x)$ может быть реализован путем коммутации входов и не требует логических элементов. Синтезируем блок $G(x)$.

Воспользуемся алгоритмом синтеза 1, подразумевающим использование только одной входной переменной для вычисления функций дополнения. Данный алгоритм дает результат, представленный в табл. 2. Наиболее простой блок $G(x)$ получается при использовании для коррекции переменной x_2 . Сложность его реализации оценивается величиной $L_{G(x)} = 640$.

Устройство, реализуемое в соответствии со структурой рис. 3, имеет сложность

$$L_{SCB_1} = L_{F(x)} + L_{H(x)} + L_{G(x)} + 23L_{XOR} + 5L_{AND} + L_{4OR} = 1080 + 560 + 640 + 23 \cdot 40 + 5 \cdot 32 + 48 = 3408.$$

Таблица 2

Описание блока контрольной логики $G(x)$ и блока $H^*(x)$ для его контроля при использовании одной переменной для коррекции

x_4	x_3	x_2	x_1	f_5	f_4	f_3	f_2	f_1	x_4					x_3				
									g_5	g_4	g_3	g_2	g_1	g_5	g_4	g_3	g_2	g_1
0	0	0	0	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1
0	0	0	1	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0
0	0	1	0	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1
0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
0	1	0	0	1	0	1	1	0	1	0	1	1	0	0	1	0	0	1
0	1	0	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	1
0	1	1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	1	0
0	1	1	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1	0
1	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0
1	0	0	1	1	0	1	0	0	0	1	0	1	1	1	0	1	0	0
1	0	1	0	1	1	0	1	0	0	0	1	0	1	1	1	0	1	0
1	0	1	1	0	1	0	1	1	1	0	1	0	0	0	1	0	1	1
1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0
1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	1	0	1	1	1	1	0	1	1	1
1	1	1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	0	1

x_2					x_1					h_4	h_3	h_2	h_1
g_5	g_4	g_3	g_2	g_1	g_5	g_4	g_3	g_2	g_1				
0	1	1	0	1	0	1	1	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0	1	1	0	0	1
0	1	0	0	0	1	0	1	1	1	1	1	0	0
1	1	1	0	1	1	1	1	0	1	0	0	1	1
1	0	1	1	0	1	0	1	1	0	1	1	0	1
0	0	1	0	0	1	1	0	1	1	0	1	1	0
0	0	1	1	0	1	1	0	0	1	0	1	0	1
0	0	1	1	0	0	0	1	1	0	0	1	0	1
1	1	0	0	0	1	1	0	0	0	0	1	0	0
1	0	1	0	0	0	1	0	1	1	1	1	1	0
0	0	1	0	1	1	1	0	1	0	0	1	1	1
1	0	1	0	0	1	0	1	0	0	1	1	1	0
0	0	1	1	1	0	0	1	1	1	0	1	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	1	0	0	0	1	1	0	0
1	1	0	0	1	1	1	0	0	1	0	1	0	0
1	1	0	0	1	1	1	0	0	1	0	1	0	1

Сложность реализации устройства по данной структуре составляет 86,939% от сложности реализации по мажоритарной структуре. Таким образом, получено улучшение показателя сложности реализации.

Синтезируем отказоустойчивое устройство в соответствии со структурой рис. 4. Блок $G(x)$ уже получен. Требуется синтез блока $H^*(x)$. В табл. 1 представлены функции данного блока. Следует отметить, что он будет одинаков для блоков $G(x)$, получаемых при коррекции сигналов по какой-либо одной из переменных, что следует из принципов построения T_m -кода. Это справедливо в силу того, что выбор той или иной переменной для коррекции приводит к инверсии всех значений функции относительно одного из вариантов дополнения либо же к сохранению значений, а для T_m -кода это не влияет на суммарный вес информационного вектора. Сложность блока $H^*(x)$ получена равной $L_{H^*(x)} = 568$. Таким образом, устройство, синтезированное по структуре рис. 4, имеет сложность реализации

$$L_{SBC_1} = L_{F(x)} + L_{G(x)} + L_{H^*(x)} + 23L_{XOR} + 5L_{AND} + L_{4OR} = 1080 + 640 + 568 + 23 \cdot 40 + 5 \cdot 32 + 48 = 34166.$$

Полученная величина несколько превышает величину сложности реализации, полученной для структуры рис. 3, – она дает устройство с 87,143% от сложности устройства, реализуемого при использовании мажоритарной структуры.

Алгоритм синтеза 2 позволяет для каждой функции дополнения выбрать наилучший вариант переменной и несколько уменьшить сложность конечного отказоустойчивого устройства. Расчеты показали, что наилучшие варианты коррекции получаются при использовании функций дополнения, вычисляемых по формулам $g_1 = f_1 \oplus x_4$, $g_2 = f_2 \oplus x_3$, $g_3 = f_3 \oplus x_4$, $g_4 = f_4 \oplus x_2$, $g_5 = f_5 \oplus x_4$. В этом случае получается блок контрольной логики со сложностью технической реализации $L_{G(x)} = 464$.

Таким образом, синтез в соответствии со структурой рис. 3, дает отказоустойчивое устройство со сложностью реализации:

$$L_{SCB_2} = L_{F(x)} + L_{H(x)} + L_{G(x)} + 23L_{XOR} + 5L_{AND} + L_{4OR} = 1080 + 560 + 464 + 23 \cdot 40 + 5 \cdot 32 + 48 = 3232.$$

Это составляет 82,449% от сложности устройства, реализованного по мажоритарной структуре.

При использовании для контроля вычислений не блока $F(x)$, а блока $G(x)$, получаем блок контрольной логики $H^*(x)$ (табл. 3) со сложностью реализации $L_{H(x)} = 560$. Отсюда следует, что устройство, синтезированное по структуре рис. 4, имеет сложность реализации

$$L_{SBC_1} = L_{F(x)} + L_{G(x)} + L_{H^*(x)} + 23L_{XOR} + 5L_{AND} + L_{4OR} = 1080 + 464 + 560 + 23 \cdot 40 + 5 \cdot 32 + 48 = 3232.$$

Это также составляет 82,449% от сложности устройства, реализованного по мажоритарной структуре.

Таблица 3

Описание блока контрольной логики $G(x)$ и блока $H^*(x)$ для его контроля при использовании различных переменных для коррекции

x_4	x_3	x_2	x_1	f_5	f_4	f_3	f_2	f_1	x_4	x_2	x_4	x_3	x_4	h_4	h_3	h_2	h_1
									g_5	g_4	g_3	g_2	g_1				
0	0	0	0	0	1	1	0	1	0	1	1	0	1	1	0	1	1
0	0	0	1	0	1	1	1	0	0	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	0	0	0	1	0	0	1	0	1	0	1	1	1	1
0	1	0	0	1	0	1	1	0	1	0	1	0	0	1	1	1	0
0	1	0	1	0	0	1	0	0	0	0	1	1	0	0	1	0	1
0	1	1	0	1	1	0	0	1	1	0	0	1	1	1	0	1	0
0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	0	1	0
1	0	0	0	1	1	0	0	0	0	1	1	0	1	1	0	1	1
1	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	1
1	0	1	0	1	1	0	1	0	0	0	1	1	1	0	1	0	0
1	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1
1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0
1	1	0	1	1	1	1	1	1	0	1	0	0	0	1	1	0	0
1	1	1	0	0	1	0	0	0	1	0	1	1	1	1	1	0	0
1	1	1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	0
1	1	1	1	0	0	1	1	0	1	1	0	0	1	0	1	0	1

Следует отметить, что структура, приведенная на рис. 4, в ряде случаев может давать даже большие значения показателей сложности технической реализации, чем структура, приведенная на рис. 3. На это влияют особенности СВК для блока логического дополнения $G(x)$: блок $H^*(x)$ может оказаться более сложным, чем аналогичный блок $H(x)$ для исходного устройства $F(x)$.

Анализ полученных результатов говорит о том, что представленные структуры дают лучшие по сложности реализации отказоустойчивые устройства, чем мажоритарная структура. Для еще большего снижения структурной избыточности следует применять для контроля вычислений менее избыточные коды, например коды с суммированием и их модификации [23, 27].

5. Экспериментальные результаты

В целях анализа эффективности применения описанных выше структур при синтезе отказоустойчивых цифровых устройств и систем были проведены эксперименты с рядом контрольных комбинационных схем из наборов LG'93 и MCNC Benchmarks [33]. Основной задачей экспериментов была оценка уменьшения показателя структурной избыточности отказоустойчивого устройства для выбранной комбинационной схемы, синтезированного по каждой из предложенных выше структур (рис. 1–4), по сравнению со структурной избыточностью мажоритарной системы (TMR, triple-modular redundancy).

В табл. 4, 5 приведены результаты экспериментов для пяти контрольных комбинационных схем $dc1$, $rd53$, $p82$, $m1$ и $max512$. В табл. 4 подробно представлены данные по показателям сложности технической реализации отказоустойчивых устройств по мажоритарной структуре и по четырем приведенным выше структурам. В качестве показателя сложности технической реализации использована площадь, занимаемая устройством на кристалле, в условных единицах библиотеки функциональных элементов *stdcell2_2.genlib*, а также относительный показатель μ – доля площади, занимаемой отказоустойчивым устройством на кристалле, от площади системы с тройной модульной избыточностью (в процентах). Площадь определялась с применением интерпретатора SIS.

При проведении экспериментов структура исходной схемы $F(x)$ не оптимизировалась (выбиралось законченное устройство). Структуры блоков $H(x)$, $G(x)$ и $H^*(x)$ оптимизировались с использованием команды *simplify* в интерпретаторе SIS. В ходе экспериментов синтезировались отказоустойчивые устройства по структуре с тройной модульной избыточностью (TMR), по обеим структурам с двой-

ной модульной избыточностью и дополнительным контролем вычислений одним из блоков (DMR, double-modular redundancy), а также по обеим структурам, использующим принцип логического дополнения (BC, Boolean complement). При этом в процессе синтеза устройств на основе структур рис. 3 и рис. 4 использовался подход, при котором корректирующие функции представляли собой одну выбранную входную переменную (см. алгоритм 1). В табл. 4 приведены данные о показателях сложности технической реализации составляющих структур, а также устройств, использующих тройную и двойную модульную избыточность, а в табл. 5 – о показателях сложности реализации структур на основе принципа логического дополнения.

Таблица 4

Показатели сложности технической реализации составляющих отказоустойчивых структур

Схема	In / Out	Площадь устройств на кристалле, усл. ед. библиотеки <i>stdcell2_2.genlib</i>										
		$F(x)$	$H(x)$	$G(x)$								
				x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
<i>dc1</i>	4/7	976	488	608	576	712	672	—	—	—	—	—
<i>rd53</i>	5/3	880	680	592	592	592	592	592	—	—	—	—
<i>p82</i>	5/14	2 368	1 784	1 976	1 792	1 920	1 864	1 848	—	—	—	—
<i>m1</i>	6/12	3 064	1 296	1 392	1 560	1 456	1 856	1 536	1 832	—	—	—
<i>max512</i>	9/6	9 632	5 480	7 832	6 984	6 456	6 872	6 696	7 104	8 352	9 376	10 192

Площадь устройств на кристалле, усл. ед. библиотеки <i>stdcell2_2.genlib</i>							μ, %
$H^*(x)$	Дополнительные элементы			Структура TMR	Структуры DMR		
	TMR	DMR	BC				
584	952	1 336	1616	3 880	3 776	97,32	
680	408	536	656	3 048	2 976	97,638	
1 784	408	536	656	7 512	7 056	93,93	
1 408	1 632	2 336	2 816	10 824	9 760	90,17	
5 232	816	1 136	1 376	29 712	25 880	87,103	

Анализируя данные таблиц, можно сделать следующие выводы. Отказоустойчивые устройства, реализуемые по предложенным в настоящей работе структурам, получаются, как правило, менее избыточными, чем отказоустойчивые устройства, реализованные по структуре с тройной модульной избыточностью и мажоритарной коррекцией сигналов. Устройства, реализуемые на основе двойной модульной избыточности с контролем вычислений одним из блоков $F(x)$ или $F^*(x)$ по T_m -коду, для большинства схем имеют выигрыш в показателе избыточности по сравнению с TMR-структурой до 10%. Для некоторых схем (например, для схемы *max512*) достигается даже большее улучшение. Реализация устройства по одной из структур, подразумевающих применение принципа логического дополнения, может еще больше сократить сложность технической реализации и уменьшить избыточность. При этом существует достаточно большая вариативность в построении структур. Например, применение алгоритма 1 при выборе функций коррекции значений позволяет строить t блоков контрольной логики (t – число входных переменных). Как видно из данных табл. 4, такая вариативность позволяет выбирать наименее избыточную реализацию блока $G(x)$ для каждого устройства $F(x)$. Это, в свою очередь, позволяет синтезировать t различных по сложности структур рис. 3 и выбирать среди них наиболее эффективный вариант. Для большинства реализаций блоков контрольной логики в рассмотренных примерах блоки $G(x)$ получались более простыми, чем блок $F(x)$, – среди 29 вариантов блоков контрольной логики только один дал блок $G(x)$ с площадью, большей чем площадь блока $F(x)$. Таким образом, практически для всех устройств применение структуры рис. 3 и принципа логического дополнения позволило уменьшить сложность технической реализации по сравнению с реализацией по структурам DMR (рис. 1, 2). Применение структуры рис. 4 в ряде случаев давало эффект в уменьшении сложности технической реализации и по сравнению со структурой рис. 3. Эффект достигался в том случае, если блок $H^*(x)$ оказывался проще блока $H(x)$. Реальный эффект от применения структуры рис. 4 замечен лишь для схемы *max512* из приведенных здесь. Для двух схем, *rd53* и

p82, сложности реализации структур рис. 3 и рис. 4 совпали. Для двух схем, $dc1$ и $m1$, структуры рис. 4 получались несколько более сложными, чем структуры рис. 3, однако превышение находилось на уровне 1–3%. На рис. 5 приведены диаграммы для сравнения показателей μ для отказоустойчивых устройств, полученных по структурам, подразумевающим использование принципа логического дополнения. Применение логического дополнения для ряда схем позволило синтезировать отказоустойчивые устройства со сложностью реализации, меньшей сложности TMR-структуры на 20–25%.

Таблица 5

**Показатели сложности технической реализации отказоустойчивых структур
на основе логического дополнения**

X1	X2	X3	X4	X5	X6	X7	X8	X9
dc1, 4/7								
Структура с контролем блока F(x)								
3 688	3 656	3 792	3 752	–	–	–	–	–
95,052	94,227	97,732	96,701					
Структура с контролем блока G(x)								
3 784	3 752	3 888	3 848	–	–	–	–	–
97,526	96,701	100,206	99,175					
rd53, 5/3								
Структура с контролем блока F(x)								
2 808	2 808	2 808	2 808	2 808	–	–	–	–
92,126	92,126	92,126	92,126	92,126				
Структура с контролем блока G(x)								
2 808	2 808	2 808	2 808	2 808	–	–	–	–
92,126	92,126	92,126	92,126	92,126				
p82, 5/14								
Структура с контролем блока F(x)								
6 784	6 600	6 728	6 672	6 656	–	–	–	–
90,309	87,859	89,563	88,818	88,605				
Структура с контролем блока G(x)								
6 784	6 600	6 728	6 672	6 656	–	–	–	–
90,309	87,859	89,563	88,818	88,605				
m1, 6/12								
Структура с контролем блока F(x)								
8 568	8 736	8 632	9 032	8 712	9 008	–	–	–
79,157	80,71	79,749	83,444	80,488	83,222			
Структура с контролем блока G(x)								
8 680	8 848	8 744	9 144	8 824	9 120	–	–	–
80,192	81,744	80,783	84,479	81,523	84,257			
max512, 9/6								
Структура с контролем блока F(x)								
24 320	23 472	22 944	23 360	23 184	23 592	24 840	25 864	26 680
81,852	78,998	77,221	78,621	78,029	79,402	83,603	87,049	89,795
Структура с контролем блока G(x)								
24 072	23 224	22 696	23 112	22 936	23 344	24 592	25 616	26 432
81,018	78,164	76,387	77,787	77,194	78,568	82,768	86,214	88,961

Следует отметить, что применение алгоритма 2 при выборе переменных для коррекции функций $f_1, f_2, \dots, f_{m-1}, f_m$, позволяет увеличить число вариантов построения блоков $G(x)$. В этом случае каждая из m функций блока контрольной логики может быть получена t способами (всего mt способов реализации). Применение алгоритма 2 может позволить еще больше улучшить показатели структурной избыточности за счет «индивидуального» подхода к каждому из рабочих выходов устройства $F(x)$.

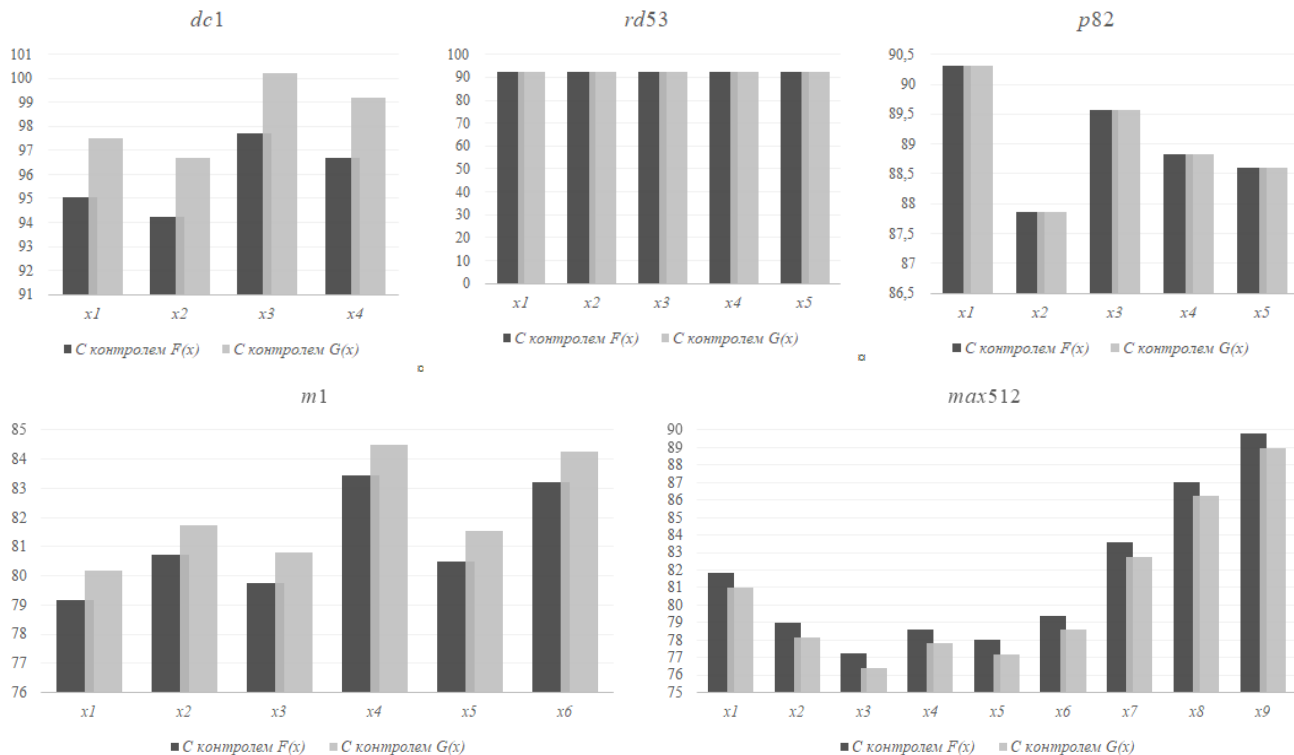


Рис. 5. Сравнение показателей структурной избыточности для различных контрольных комбинационных схем
 Fig. 5. Comparison of indicators of structural redundancy for various combinational benchmarks

Заключение

При построении отказоустойчивых цифровых вычислительных устройств и систем могут быть использованы не только традиционные подходы, основанные на тройной модульной избыточности с мажоритарной коррекцией сигналов и двойной модульной избыточности с применением схем сравнения и переключения, но и подходы, основанные на использовании СВК отдельных блоков и логического дополнения для фиксации искаженных сигналов. Подобные структуры синтезируются с учетом следующего ограничения: одновременно возможно возникновение неисправностей только в одном из блоков системы.

Структуры отказоустойчивых устройств и систем, основанные на использовании двойной модульной избыточности с контролем вычислений одним из блоков (основным или резервным) по T_m -коду, для многих реальных устройств $F(x)$ позволяют получать менее избыточные отказоустойчивые реализации, чем применение мажоритарной структуры. При этом данные структуры обладают высокой корректирующей способностью за счет возможности фиксации любых сочетаний искажений на выходах объекта диагностирования, за исключением искажений с кратностями $d = m$. Применение принципа логического дополнения при синтезе блока ФИС позволяет еще больше сократить структурную избыточность конечного отказоустойчивого устройства, что объясняется возможностями существенного упрощения блока логического дополнения $G(x)$.

Необходимо отметить две особенности описанных в настоящей статье структур. Во-первых, за счет использования нескольких последовательных каскадов сумматоров по модулю $M = 2$ во всех предложенных структурах примерно вдвое уменьшено быстродействие по сравнению с традиционными структурами (пути прохождения сигналов к выходному каскаду увеличены). Однако при современных технологиях реализации цифровых устройств и систем такое снижение быстродействия не представляется столь существенным. Во-вторых, если использование принципов модульной избыточности с устройствами сравнения и коррекции позволяет не анализировать структуры исходных объектов $F(x)$ и строить их из типовых блоков, то в случае применения описанных подходов такой

анализ требуется при построении СВК и блока ФИС. Для устройств с большим числом входов (более 30) данная задача может быть достаточно трудоемкой, что потребует применения методов декомпозиции при синтезе соответствующих устройств и более сложных процедур проектирования отказоустойчивых устройств и систем. Тем не менее при автоматизации процедур получения функций логического дополнения можно добиться существенного упрощения конечных отказоустойчивых устройств и систем, реализуемых на современной элементной базе, по сравнению с традиционными подходами для решения этой задачи.

Применение логического дополнения и кодовых методов для синтеза самопроверяемых и отказоустойчивых цифровых устройств и систем позволяет расширить число способов их реализации и минимизировать показатели структурной избыточности. Полученные в статье результаты целесообразно учитывать на практике при выборе подхода к синтезу отказоустойчивых цифровых устройств и систем.

В заключение также обратим внимание на возможные перспективы в продолжении исследований в области синтеза отказоустойчивых устройств и систем. Прежде всего отметим еще раз, что предложенные в настоящей статье структуры ориентированы на работу с одиночными неисправностями. Тем не менее при использовании подобных структур в ряде приложений, например в составе устройств и систем критического применения, где входные воздействия могут изменяться достаточно редко [34], будут создаваться условия накопления скрытых неисправностей. В аварийном режиме скрытые неисправности могут начать проявляться в виде кратных отказов, а их последствия – в виде искажения вычисляемых функций, в конечном итоге не корректироваться в предложенных структурах. В дальнейших исследованиях необходимо обращать внимание на проблему контролепригодности подобных структур при их функционировании в штатном режиме и на возможности обнаружения искажаемых функций в рабочем режиме и локализации скрытых дефектов [35–37]. Ее решение может быть связано с повышением обнаруживающей способности схемы контроля, а также с организацией систем тестово-функционального диагностирования [38].

Список источников

1. Щербаков Н.С. Самокорректирующиеся дискретные устройства. М. : Машиностроение, 1975. 216 с.
2. Согомонян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М. : Радио и связь, 1989. 208 с.
3. Lala P.K. Self-Checking and Fault-Tolerant Digital Design. San Francisco : Morgan Kaufmann Publishers, 2001. 216 p.
4. Sogomonyan E.S. Self-Correction Fault-Tolerant Systems : Preprint. 2018, 30 p. URL: https://www.researchgate.net/publication/328355722_Self-Correction_Fault-Tolerant_Systems
5. Sellers F.F., Hsiao M.-Y., Beamson L.W. Error Detecting Logic for Digital Computers. New York : McGraw-Hill, 1968, XXI + 295 p.
6. Гаврилов М.А., Остиану В.М., Потехин А.И. Надежность дискретных систем. М. : 1970. 104 с. (Итоги науки и техники. Сер. Теория вероятностей. Математическая статистика. Теоретическая кибернетика).
7. Сапожников В.В., Сапожников Вл.В., Христов Х.А., Гавзов Д.В. Методы построения безопасных микроэлектронных систем железнодорожной автоматики / под ред. Вл.В. Сапожникова. М. : Транспорт, 1995. 272 с.
8. Matrosova A.Yu., Levin I., Ostanin S.A. Self-Checking Synchronous FSM Network Design with Low Overhead // VLSI Design. 2000. V. 11, is. 1. P. 47–58.
9. Сляяр В.В., Харченко В.С. Отказоустойчивые компьютерные системы управления с версионно-пороговой адаптацией: способы адаптации, оценка надежности, выбор архитектур // Автоматика и телемеханика. 2002. № 6. С. 131–145.
10. Останин С.А. Синтез отказоустойчивых комбинационных схем // Прикладная дискретная математика (приложение № 1). 2009. № 1. С. 71–72.
11. Hamamatsu M., Tsuchiya T., Kikuno T. Finding the Optimal Configuration of a Cascading TMR System // 14th IEEE Pacific Rim International Symposium on Dependable Computing, 15–17 December 2008, Taipei, Taiwan. P. 329–350.
12. Matsumoto K., Uehara M., Mori H. Evaluating the Fault Tolerance of Stateful TMR // 13th International Conference on Network-Based Information Systems, 14–16 September 2010, Takayama, Japan. P. 332–336.
13. Бочков К.А., Харлап С.Н., Сивко Б.В. Разработка отказоустойчивых систем на основе диверситетных базисов // Автоматика на транспорте. 2016. Т. 2, № 1. С. 47–64.
14. Стемповский А.Л., Тельпухов Д.В., Жукова Т.Д., Гуров С.И., Соловьев Р.А. Методы синтеза сбоеустойчивых комбинационных КМОП схем, обеспечивающих автоматическое исправление ошибок // Известия ЮФУ. Технические науки. 2017. № 7 (192). С. 197–210.
15. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Схема коррекции сигналов для комбинационных устройств автоматики на основе логического дополнения с контролем вычислений по паритету // Информатика. 2020. Т. 17, № 2. С. 71–85.

16. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Отказоустойчивая структура на основе логического дополнения с контролем вычислений по паритету // Автоматика на транспорте. 2020. Т. 6, № 3. С. 377–403.
17. Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В. Логическое дополнение – новый метод контроля комбинационных схем // Автоматика и телемеханика. 2003. № 1. С. 167–176.
18. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking. Ed. 1. Dordrecht : Springer Science + Business Media B.V., 2008. 184 p.
19. Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M. Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proc. of the 10th International Workshops on Boolean Problems, Freiberg, Germany, September, 2012. P. 33–40.
20. Saposhnikov V.I., Dmitriev A., Goessel M., Saposhnikov V.V. Self-Dual Parity Checking – a New Method for on Line Testing // Proc. of 14th IEEE VLSI Test Symposium. USA, Princeton, 1996. P. 162–168.
21. Efanov D., Sapozhnikov V., Sapozhnikov V.I., Osadchy G., Pivovarov D. Self-Dual Complement Method up to Constant-Weight Codes for Arrangement of Combinational Logical Circuits Concurrent Error-Detection Systems // Proc. of 17th IEEE East-West Design & Test Symposium (EWDTS'2019), Batumi, Georgia, September 13–16, 2019. P. 136–143.
22. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Дмитриев В.В. Новые структуры систем функционального контроля логических схем // Автоматика и телемеханика. 2017. № 2. С. 127–143.
23. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Коды с суммированием для систем технического диагностирования. М. : Наука, 2021. Т. 2: Взвешенные коды с суммированием. 455 с.
24. Sapozhnikov V., Sapozhnikov V.I., Efanov D. Typical Signal Correction Structures Based on Duplication with the Integrated Control Circuit // Proc. of 18th IEEE East-West Design & Test Symposium (EWDTS'2020), Varna, Bulgaria, September 4–7, 2020. P. 78–87.
25. Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I. Typical Structure of a Duplicate Error Correction Scheme with Code Control with Summation of Weighted Transitions // Electronic Modeling. 2020. V. 42, is. 5. P. 38–50.
26. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Коды Хэмминга в системах функционального контроля логических устройств. СПб. : Наука, 2018. 151 с.
27. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Коды с суммированием для систем технического диагностирования. М. : Наука, 2020. Т. 1: Классические коды Бергера и их модификации. 383 с.
28. Nikolos D. Self-Testing Embedded Two-Rail Checkers // Journal of Electronic Testing. 1998. V. 12. P. 69–79.
29. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Основы теории надежности и технической диагностики. СПб. : Лань, 2019. 588 с.
30. Zakrevskij A., Pottosin Yu., Cheremisinova L. Optimization in Boolean Space. Tallinn : TUT Press, 2009. 241 p.
31. Sentovich E.M., Singh K.J., Moon C., Savoj H., Brayton R.K., Sangiovanni-Vincentelli A. Sequential Circuit Design Using Synthesis and Optimization // Proc. IEEE International Conference on Computer Design: VLSI in Computers & Processors, 11–14 October 1992, Cambridge, MA, USA. P. 328–333.
32. SIS: A System for Sequential Circuit Synthesis / E.M. Sentovich, K.J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P.R. Stephan, R.K. Brayton, A. Sangiovanni-Vincentelli / Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California. Berkeley, 1992. 45 p.
33. Collection of Digital Design Benchmarks. URL: <http://ddd.fit.cvut.cz/www/prj/Benchmarks/>
34. Сапожников Вл.В. Синтез систем управления движением поездов на железнодорожных станциях с исключением опасных отказов. М. : Наука, 2021. 230 с.
35. Drozd A., Kharchenko V., Antoshchuk S., Sulima J., Drozd M. Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions // Proc. of 9th IEEE East-West Design & Test Symposium (EWDTS'2011). Sevastopol, 2011. P. 411–416.
36. Drozd A., Drozd M., Martynyuk O., Kuznietsov M. Improving of a Circuit Checkability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems // CEUR Workshop Proc. 2017. Vol. 1844. P. 654–661. URL: <http://ceur-ws.org/Vol-1844/10000654.pdf>
37. Drozd O., Perebeinos I., Martynyuk O., Zashcholkina K., Ivanova O., Drozd M. Hidden Fault Analysis of FPGA Projects for Critical Applications // Proc. of the IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2020, 25–29 February, Lviv-Slavsko, Ukraine. Paper 142.
38. Литиков И.П., Согомонян Е.С. Тестово-функциональное диагностирование цифровых устройств и систем // Автоматика и телемеханика. 1985. № 3. С. 111–121.

References

1. Shcherbakov, N.S. (1975) *Samokorrektiruyushchiesya diskretnye ustroystva* [Self-correcting discrete devices]. Moscow: Mashinostroenie.
2. Sogomonyan, E.S. & Slabakov, E.V. (1989) *Samoproveryaemye ustroystva i otkazoustoychivye sistemy* [Self-checking devices and fault-tolerant systems]. Moscow: Radio i svyaz'.
3. Lala, P.K. (2001) *Self-checking and fault-tolerant digital design*. San Francisco: Morgan Kaufmann Publishers.
4. Sogomonyan, E.S. (2018) *Self-correction fault-tolerant systems*. [preprint].
5. Sellers, F.F., Hsiao, M.-Y. & Beamson, L.W. (1968) *Error Detecting Logic For Digital Computers*. New York: McGraw-Hill.

6. Gavrilov, M.A., Ostianu, V.M. & Potekhin, A.I. (1969, 1970) *Nadezhnost' diskretnykh sistem* [Reliability of Discrete Systems]. Moscow: [s.n.]. pp. 7–104.
7. Sapozhnikov, V.V., Sapozhnikov, V.I., Khristov, H.A. & Gavzov, D.V. (1995) *Metody postroeniya bezopasnykh mikroelektronnykh sistem zheleznodorozhnoy avtomatiki* [Methods for constructing safety microelectronic systems of railway automation]. Moscow: Transport.
8. Matrosova, A.Yu., Levin I. & Ostanin, S.A. (2000) Self-Checking Synchronous FSM Network Design with Low Overhead. *VLSI Design*. 11(1). pp. 47–58. DOI: 10.1155/2000/46578.
9. Sklyar, V.V. & Kharchenko, V.S. (2002) Fault-Tolerant Computer-Aided Control Systems with Multiversion-Threshold Adaptation: Adaptation Methods, Reliability Estimation, and Choice of an Architecture. *Avtomatika i telemekhanika – Automation and Remote Control*. 6. pp. 131–145.
10. Ostanin, S.A. (2009) Sintez otказoustoychivyykh kombinatsionnykh skhem [Synthesis of Fault-Tolerant Combinational Circuits]. *Prikladnaya diskretnaya matematika (prilozhenie № 1)*. 1. pp. 71–72.
11. Hamamatsu, M., Tsuchiya, T. & Kikuno, T. (2008) Finding the Optimal Configuration of a Cascading TMR System. *14th IEEE Pacific Rim International Symposium on Dependable Computing*. Taipei, Taiwan. December 15–17, 2008. pp. 329–350. DOI: 10.1109/PRDC.2008.12.
12. Matsumoto, K., Uehara, M. & Mori, H. (2010) Evaluating the Fault Tolerance of Stateful TMR. *13th International Conference on Network-Based Information Systems*. Takayama, Japan. September 14–16, 2010. pp. 332–336. DOI: 10.1109/NBiS.2010.86
13. Bochkov, K.A., Harlap, S.N. & Sivko, B.V. (2016) Design of Axiomatic Based Fault-Tolerant Systems. *Avtomatika na transporte – Automation on Transport*. vol. 2, Issue 1, pp. 47–64.
14. Stempkovskiy, A.L., Telpuhov, D.V., Zhukova, T.D., Gurov, S.I. & Soloviev, R.A. (2017) Synthesis Methods of Fault-Tolerant Combination CMOS Circuits, Providing Automatic Correction of Errors. *Izvestiya YuFU. Tekhnicheskie nauki – Izvestiya SFedU. Engineering sciences*. 7(192). pp. 197–210. DOI: 10.23683/2311-3103-2017-7-197-210.
15. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2020) Signal Correction for Combinational Automation Devices on the Basis of Boolean Complement with Control of Calculations by Parity. *Informatika – Informatics*. 17(2). pp. 71–85. DOI: 10.37661/1816-0301-2020-17-2-71-85.
16. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2020) Fault-Tolerant Structure Based on Boolean Complement with Parity Calculations Control. *Avtomatika na transporte – Automation on Transport*. 6(3). pp. 377–403. DOI: 10.20295/2412-9186-2020-6-3-377-403.
17. Gessel, M., Morozov, A.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2003) Logic Complement, a New Method of Checking the Combinational Circuits. *Avtomatika i telemekhanika – Automation and Remote Control*. 1. pp. 167–176.
18. Gössel, M., Ocheretny, V., Sogomonyan, E. & Marienfeld, D. (2008) *New Methods of Concurrent Checking*. 1st ed. Dordrecht: Springer Science+Business Media B.V.
19. Das, D.K., Roy, S.S., Dmitriev, A., Morozov, A. & Gössel, M. (2012) Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes. *Proc. of the 10th International Workshops on Boolean Problems*. Freiberg, Germany, September. pp. 33–40.
20. Sapozhnikov, V.I., Dmitriev, A., Gössel, M. & Sapozhnikov, V.V. (1996) Self-Dual Parity Checking – a New Method for on Line Testing. *Proceedings of 14th IEEE VLSI Test Symposium*. USA, Princeton. pp. 162–168.
21. Efanov, D., Sapozhnikov, V., Sapozhnikov, V.I., Osadchy, G. & Pivovarov, D. (2019) Self-Dual Complement Method up to Constant-Weight Codes for Arrangement of Combinational Logical Circuits Concurrent Error-Detection Systems, *Proc. of 17th IEEE East-West Design & Test Symposium (EWDTS'2019)*. Batumi, Georgia. September 13–16. pp. 136–143. DOI: 10.1109/EWDTS.2019.8884398
22. Sapozhnikov, V.V., Sapozhnikov, V.I., Efanov, D.V. & Dmitriev, V.V. (2017) New Structures of the Concurrent Error Detection Systems for Logic Circuits. *Automation and Remote Control*. 2. pp. 127–143.
23. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2021) *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya* [Sum codes for technical diagnostics systems]. Vol. 2. Moscow: Nauka.
24. Sapozhnikov, V., Sapozhnikov, V.I. & Efanov, D. (2020) Typical Signal Correction Structures Based on Duplication with the Integrated Control Circuit. *Proc. of 18th IEEE East-West Design & Test Symposium (EWDTS'2020)*. Varna, Bulgaria. September 4–7, 2020. pp. 78–87. DOI: 10.1109/EWDTS50664.2020.9224649
25. Efanov, D.V., Sapozhnikov, V.V. & Sapozhnikov, V.I. (2020) Typical Structure of a Duplicate Error Correction Scheme with Code Control with Summation of Weighted Transitions. *Electronic Modeling*. 42(5). pp. 38–50. DOI: 10.15407/emodel.42.05.038
26. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2018) *Kody Khemminga v sistemakh funktsional'nogo kontrolya logicheskikh ustroystv* [Hamming codes in concurrent error detection systems of logic devices]. St. Petersburg: Nauka.
27. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2020) *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya* [Sum Codes for Technical Diagnostics Systems]. Vol. 1. Moscow: Nauka.
28. Nikolos, D. (1998) Self-Testing Embedded Two-Rail Checkers. *Journal of Electronic Testing*. 12. pp. 69–79. DOI: 10.1007/978-1-4757-60-69-9_7.
29. Sapozhnikov, V.V., Sapozhnikov, V.I. & Efanov, D.V. (2019) *Osnovy teorii nadezhnosti i tekhnicheskoy diagnostiki* [Fundamentals of the theory of reliability and technical diagnostics]. St. Petersburg: Lan'.
30. Zakrevskiy, A., Pottosin, Yu. & Cheremisinova, L. (2009) *Optimization in Boolean Space*. Tallinn: TUT Press.

31. Sentovich, E.M., Singh, K.J., Moon, C., Savoj, H., Brayton, R.K. & Sangiovanni-Vincentelli, A. (1992) Sequential Circuit Design Using Synthesis and Optimization. *Proc. IEEE International Conference on Computer Design: VLSI in Computers & Processors*. Cambridge, MA, USA. October 11–14. pp. 328–333. DOI: 10.1109/ICCD.1992.276282
32. Sentovich, E.M., Singh, K.J., Lavagno, L., Moon, C., Murgai, R., Saldanha, A., Savoj, H., Stephan, P.R., Brayton, R.K. & Sangiovanni-Vincentelli, A. (1992) *SIS: A system for sequential circuit synthesis. electronics research laboratory*. Department of Electrical Engineering and Computer Science. University of California, Berkeley. 4th May.
33. Czechia. (n.d.) Collection of Digital Design Benchmarks. [Online] Available from: <http://ddd.fit.cvut.cz/www/prj/Benchmarks/>
34. Sapozhnikov, V.I.V. (2021) *Sintez sistem upravleniya dvizheniem poezdov na zheleznodorozhnykh stantsiyakh s isklucheniem opasnykh otkazov* [Synthesis of train traffic control systems at railway stations with the exception of dangerous failures]. Moscow: Nauka.
35. Drozd, A., Kharchenko, V., Antoshchuk, S., Sulima, J. & Drozd, M. (2011) Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions. *Proc. of 9th IEEE East-West Design & Test Symposium (EWDTS'2011)*. Sevastopol, Ukraine. pp. 411–416. DOI: 10.1109/EWDTS.2011.6116606
36. Drozd, A., Drozd, M., Martynyuk, O. & Kuznietsov, M. (2017) Improving of a Circuit Checkability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems. *CEUR Workshop Proceedings*. 1844. pp. 654–661. [Online] Available from: <http://ceur-ws.org/Vol-1844/10000654.pdf>
37. Drozd, O., Perebeinos, I., Martynyuk, O., Zashcholkina, K., Ivanova, O. & Drozd, M. (2020) Hidden Fault Analysis of FPGA Projects for Critical Applications. *Proc. of the IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*. Lviv-Slavsko, Ukraine. February 25–29. Paper 142. DOI: 10.1109/TCSET49122.2020.235591
38. Litikov, I.P. & Sogomonyan, E.S. (1985) Test and Functional Diagnosis of Digital Devices and Systems. *Avtomatika i telemekhanika – Automation and Remote Control*. 3. pp. 111–121.

Информация об авторе:

Ефанов Дмитрий Викторович – доктор технических наук, профессор кафедры автоматизации, телемеханики и связи на железнодорожном транспорте Российского университета транспорта (Москва, Россия); профессор Высшей школы транспорта Института машиностроения, материалов и транспорта Санкт-Петербургского политехнического университета Петра Великого (Санкт-Петербург, Россия). E-mail: TrES-4b@yandex.ru

Information about the author:

Efanov Dmitry Viktorovich – Doctor of Technical Sciences, Professor, Russian University of Transport (Moscow, Russian Federation); Peter the Great St. Petersburg Polytechnic University (St. Petersburg, Russian Federation). E-mail: TrES-4b@yandex.ru

Поступила в редакцию 11.06.2021; принята к публикации 28.02.2022

Received 11.06.2021; accepted for publication 28.02.2022