УДК 681.03

В.И. Барсов[1], Е.А. Сотник[1], В.А. Краснобаев[2]

[1] *Украинская инженерно-педагогическая академия, Харьков*
[2] *Полтавский национальный технический университет им. Ю. Кондратюка, Полтава*

ИССЛЕДОВАНИЕ МЕТОДОВ КОНТРОЛЯ И ДИАГНОСТИКИ ОШИБОК СПЕЦПРОЦЕССОРА ОБРАБОТКИ ИНФОРМАЦИИ, ФУНКЦИОНИРУЮЩЕГО НА ОСНОВЕ КОДОВ МОДУЛЯРНОЙ СИСТЕМЫ СЧИСЛЕНИЯ

Рассмотрены и проанализированы достоинства и недостатки известных методов контроля и диагностики ошибок СПОИ, функционирующего в МСС. Разработан и исследован метод параллельной коррекции, способный повысить эффективность контроля и диагностики ошибок, независимо от вида вычислительной цепи и особенностей распределения ошибки в ней.

Ключевые слова: *модулярная система счисления, метод стягивания альтернативных совокупностей, метод принятия гипотезы об ошибочном основании, метод параллельной коррекции, спецпроцессор обработки информации.*

Введение

На сегодняшний день существует и используется два основных метода контроля и диагностики ошибок спецпроцессора функционирующего в модулярной системе счисления (МСС) [1]:

Первый метод заключается в последовательных определениях условных альтернативных совокупностей (АС) в ходе выполнения программы и их стягивании к ошибочному основанию. Условной альтернативной совокупностью $\tilde{W}(\tilde{A})$ неправильного числа $\tilde{A}$ называется совокупность оснований, по которым возможна ошибка, с учетом характера альтернативных совокупностей предшествующих неправильных результатов по ходу выполнения программы.

Исходя из этого, если в упорядоченной системе оснований $m_1, m_2, ..., m_n, m_{n+1}$ задано неправильное число $\tilde{A}$, имеющее альтернативную совокупность $W(\tilde{A}) = (m_{i1}, m_{i2}, ..., m_{ik})$, и если при выполнении над числом $\tilde{A}$ рациональной операции по программе было получено неправильное число $\tilde{B}$, альтернативная совокупность которого равна $W(\tilde{B}) = (\bar{m}_{i1}, \bar{m}_{i2}, ..., \bar{m}_{ik})$, то ошибочной может быть цифра по какому-либо из оснований

$$(m_{j1}, m_{j2}, ..., m_{jk}) = W(\tilde{A})W(\tilde{B}),$$

где умножение понимается в смысле пересечения.

Это логично, поскольку среди альтернативных совокупностей $W(\tilde{A})$ и $W(\tilde{B})$ всегда содержится ошибочное основание, а значит если в $W(\tilde{A})$ есть такие основания, которых нет в $W(\tilde{B})$ и наоборот, то, очевидно, что среди них нет ошибочного. Таким образом процесс обнаружения ошибки требует определения всех проекций числа после каждого действия над неправильным числом $\tilde{A}$. Кроме того контрольное основание всегда будет входить в УАС и, соответственно, после стягивания к двум ошибочным основаниям приходится делать допущение о его истинности, что в свою очередь не дает гарантии правильного исправления ошибки.

На рис. 1 – 3 представлены результаты экспериментальной оценки скорости стягивания к ошибочному основанию методом стягивания. Метод моделировался для трех вариантов вычислительной цепи. I – когда операции сложения и умножения соотносятся как 80%/20%; II – 50/50%; III – 20%/80%. Анализ показал, что для I варианта вычислительной цепи требуется не менее пяти действие до выявления ошибки. Для II только при ошибке по контрольному основанию стягивание реализуется за три вычислительных операции с ошибочным числом. Еще по двум основаниям необходимо пять операций. Для всех остальных требуется девять и более арифметических действий с неправильным числом. В III варианте по двум основаниям выявление ошибки производится уже после первого действия, по седьмому основанию достаточно трех действий. Во всех остальных случаях требуется восемь и более действий. Это позволяет сделать вывод, что метод стягивания не эффективен в коротких вычислительных цепях.

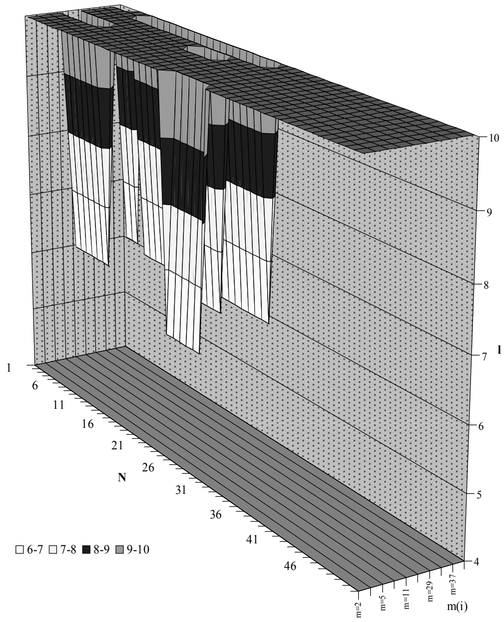

Рис. 1. Длина цепи операций до обнаружения места ошибки методом стягивания для I варианта вычислительно цепи

\

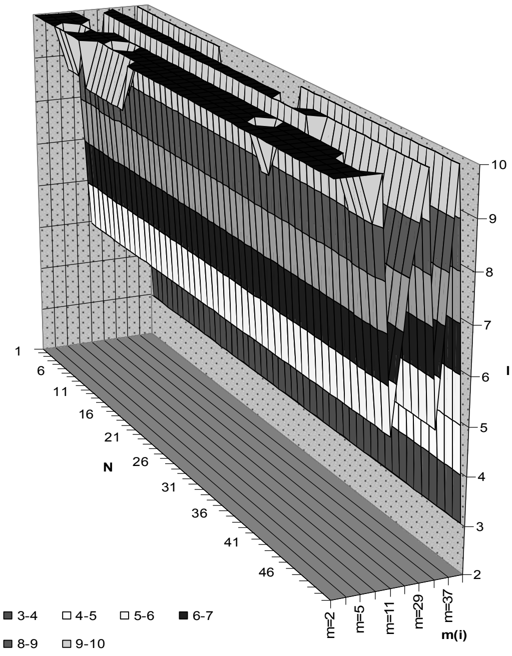

Рис. 2. Длина цепи операций до обнаружения места ошибки методом стягивания для II варианта вычислительно цепи

Второй метод состоит в принятии какой-либо гипотезы по альтернативной совокупности числа $\tilde{A}$, проведении коррекции по этой гипотезе и выполнении дальнейших вычислений вплоть до обнаружения правильности или несостоятельности принятой гипотезы.

В основе метода гипотез лежит следующее утверждение.

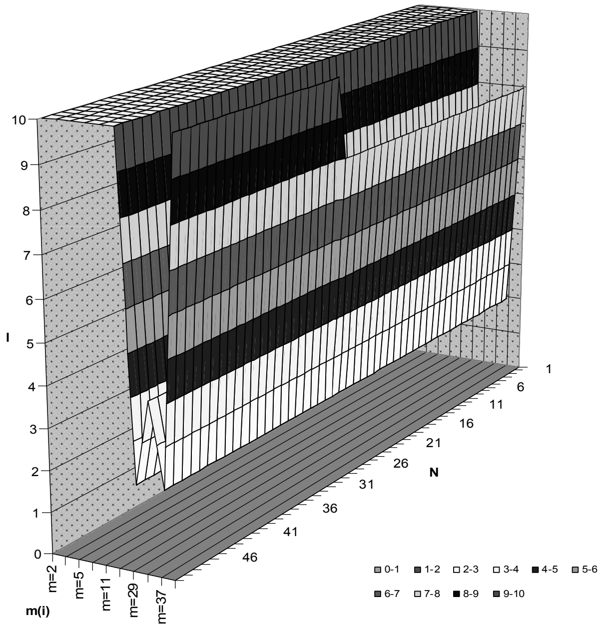

Рис. 3. Длина цепи операций до обнаружения места ошибки методом стягивания для III варианта вычислительно цепи

Пусть в упорядоченной системе оснований $m_1, m_2, ..., m_n, m_{n+1}$ задано неправильное число $\tilde{A}$, с альтернативной совокупностью $W(\tilde{A})$, в которую входит основание m_j и задано, что в случае принятия гипотезы неправильности цифры a'_j по основанию m_j последняя должна исправиться на a_j. Тогда, если при вычислении некоторой рациональной функции f($\tilde{A}$ *)*

$$\tilde{C} = f(\tilde{A}) = (\gamma'_1, \gamma'_2, ..., \gamma'_j, ..., \gamma'_n, \gamma'_{n+1}),$$

такой, что

$$\gamma'_j = f(a'_j) = f(a_j),$$

получаем в результате неправильное число $\tilde{C}$, то цифра a'_j не может быть ошибочной. Действительно, если цифра a'_j ошибочна, то правильной является цифра a_j. Так как при вычислении $\tilde{C}$ цифра γ_j получается такой же, как если бы в $\tilde{A}$ по основанию m_j была правильная цифра a_j, то $\tilde{C}$ должно быть правильным числом. Соответственно факт неправильности $\tilde{C}$ опровергает предположение об ошибочности цифры a'_j.

Данный метод позволяет производить значительно более эффективный контроль и диагностику в сравнении с методом стягивания, что доказывают результаты эксперимента, представленные на рис. 4 – 6. Как видно, скорость определения ошибочной цифры напрямую зависит от основания, по которому она произошла.

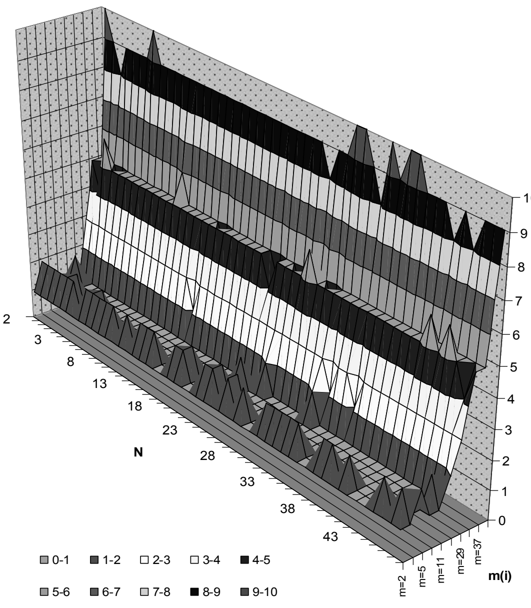

Рис. 4. Длина цепи операций до обнаружения места ошибки методом принятия гипотезы для I варианта вычислительно цепи

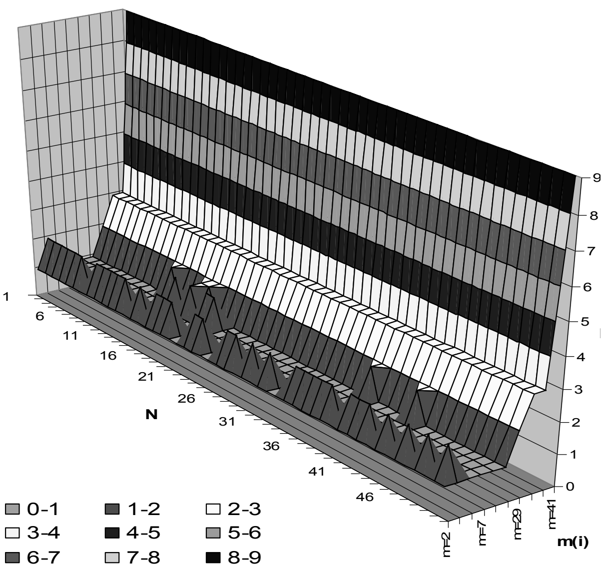

Рис. 6. Длина цепи операций до обнаружения места ошибки методом принятия гипотезы для III варианта вычислительно цепи

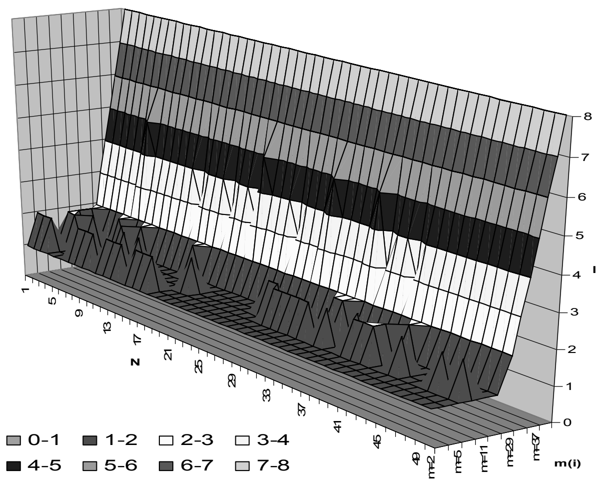

Рис. 5. Длина цепи операций до обнаружения места ошибки методом принятия гипотезы для II варианта вычислительно цепи

Так для всех вариантов вычислительной цепи на малых основаниях ошибка диагностируется уже после первого (реже после второго) действия. Однако на больших основаниях длительность определения места ошибки увеличивается до трех – пяти (для последнего информационного основания до девяти) арифметических операций. Данный факт позволяет утверждать, что применение метода принятия гипотезы обосновано для вычислительной цепи где вероятность появления ошибки по большим основаниям системы значительно меньше чем по малым. Кроме того значительным недостатком данного метода является необходимость возвращаться назад в вычислительной цепи в случае принятия неверной гипотезы.

Данный факт не позволяет организовать эффективные контроль и диагностику в динамике вычислительных действий.

Можно заключить, что данные методы не способны одинаково эффективно выполнять функции контроля и диагностики ошибок в динамике вычислительного процесса в различных вычислительных цепях с произвольными вероятностями распределения ошибок. Поэтому актуальным является поиск решения данной научно-технической задачи позволяющего повысить эффективность функционирования спецпроцессора обработки информации (СПОИ) в динамике вычислений.

Основная часть

Отличительной способностью кодов в остаточных классах является возможность эффективного контроля и диагностики ошибок в динамике вычислений. Под «динамикой» будем подразумевать обнаружение наличия ошибки и определение ее места без остановки вычислений либо возврата к уже пройденным этапам вычислительной цепи. Использование для этой цели кодовых слов с двумя контрольными основаниями повышает вероятность обнаружения и исправления ошибок, однако создает излишнюю структурную избыточность. В случаях, когда в конкретном СПОИ, ввиду его архитектурных особенностей, вероятность возникновения двойных и более ошибок ничтожно мала, целесообразным является использование только одного контрольного основания. Многочисленные исследования МСС [1 – 3] показали, что коды с одним контрольным основанием гарантированно дают возможность обнаружить однократную ошибку и в 90%

случаев ее исправить. Как уже было показано, основной принцип существующих методов определения ошибочного остатка a_j состоит в том, что для получаемой в результате операций совокупности неправильных операндов $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_\rho$, в процессе выполнения программы последовательно во времени определяются условные альтернативные совокупности (УАС) вида

$$W\left(\tilde{A}\right) = W\left(\tilde{A}_1\right) \wedge W\left(\tilde{A}_2\right) \wedge \ldots \wedge W\left(\tilde{A}_\rho\right),$$

где $W\left(\tilde{A}_\ell\right) = \left\{m_{q_1}, m_{q_1}, \ldots, m_{q_{i_\rho}}\right\}$ – альтернативная совокупность ℓ-го неправильного числа. Либо по первой альтернативной совокупности выдвигается допущение о неправильности одного из оснований, которое может не подтвердиться. В первом случае вероятность сведения $W(\tilde{A})$ к одному основанию возможно, только если ошибка произошла по контрольному основанию, соответственно в большинстве случаев $W(\tilde{A}) = (m_i, m_n+1)$, что оставляет вероятность неверной коррекции. Во втором случае при постановке неверной гипотезы возникает необходимость остановить вычислительный процесс и вернуться к месту возникновения ошибки для повторной коррекции цифры по новому основанию, входящему в $W(\tilde{A})$. Это противоречит поставленному нами условию контроля и диагностики в динамике вычислительного процесса. Данные недостатки являются той причиной, по которой в кодах с одним контрольным основанием возможна коррекция лишь 90% ошибок. Таким образом, необходимы и актуальны исследования, посвященные разработке метода, который сохранил бы в себе достоинства существующих методов стягивания и принятия гипотез, и при этом был бы нечувствительным к рассмотренным недостаткам.

С этой целью авторами был разработан метод параллельной коррекции, суть которого заключается в следующем. После каждого действия в вычислительной цепи производится контроль наличия ошибки в числе. Т.е. определяется, удовлетворяет ли число А условию

$$A < \frac{M_1}{m_{n+1}}, \tag{1}$$

где $M_1 = M \cdot m_{n+1}$ – полный диапазон системы, $M = m_1 \cdot m_2 \cdot \ldots \cdot m_n$ – рабочий диапазон

Если после передачи число А не удовлетворяет условию (1), то оно является неправильным и производится определение альтернативной совокупности искаженного числа $\tilde{A}$.

$$W(\tilde{A}) = \left(m_{i_1}, m_{i_2}, \ldots, \tilde{m}_{i_j}, \ldots, m_{i_n}\right). \tag{2}$$

Среди оснований, входящих в альтернативную совокупность в (2) и учитывая условие, что возможна только единичная ошибка, обязательно присутствует ошибочное $\tilde{m}_{i_j}$, причем только одно.

В цифрах по всем основаниям, входящим в АС $W(\tilde{A})$ в (2), независимо от остальных, известными методами производится коррекция. Т.е. при коррекции цифры по основанию m_{i_1} делается допущение, что все остальные цифры по основаниям, входящим в $W(\tilde{A})$ - верны. Однако скорректированное значение цифры a'_{i_1} по основанию m_{i_1} не заменяет текущее значения цифры в числе $\tilde{A}$, а сохраняется в отдельной ячейке памяти. Те же действия повторяются для всех остальных оснований входящих в $W(\tilde{A})$.

Таким образом, числовое значение числа $\tilde{A}$ остается неизменным и ошибочным. Фактически будет сохранено число $\tilde{A}$

$$\tilde{A} = (a_1, a_2, \ldots, \tilde{a}_i, \ldots, a_n) \tag{3}$$

и отдельно, но с учетом (3), скорректированные по своим основаниям цифры.

$$\begin{aligned} A'_{i1} &= (a'_{i1}) \bmod m_{i1}; \\ A'_{i2} &= (a'_{i2}) \bmod m_{i2}; \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ A'_{i_j} &= (a'_{i_j}) \bmod m_{i_j}; \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ A'_{i_n} &= (a'_{i_n}) \bmod m_{i_n}. \end{aligned} \tag{4}$$

Так как во время коррекции для определения каждого значения A'_{i_1} за основу бралось число $\tilde{A}$ (3), в котором исправлялась только одна цифра и само число не изменялось, то в (4) одно значение

$$A'_{i_j} = (a'_{i_j}) \bmod m_{i_j}$$

содержит истинное значение цифры a_i, числа А, тогда как все остальные значения из истинных стали неправильными.

Учитывая, что $A'_{i_1}, A'_{i_2}, \ldots, A'_{i_n}$ используют те же основания что и число А, то не возникает никакой структурной избыточности, как это было бы с двумя контрольными основаниями.

После выполнения параллельной коррекции выполняется следующая запланированная в вычислительной цепи операция над числом $\tilde{A}$ и полученными скорректированными числами $A'_{i_1}, A'_{i_2}, \ldots, A'_{i_n}$.

Запишем это, учитывая (3, 4).

$$\begin{aligned} C &= \tilde{A} \otimes B = \\ &= (a_1 \otimes b_1, a_2 \otimes b_2, \ldots, \tilde{a}_i \otimes b_i, \ldots, a_n \otimes b_n); \end{aligned} \tag{5}$$

$$c_{k_1} = A'_{i_1} \otimes b_{i_1} = \left(a'_{i_1} \otimes b_{i_1}\right) \bmod m_{i_1};$$

$$c_{k_2} = A'_{i_2} \otimes b_{i_2} = \left(a'_{i_2} \otimes b_{i_2}\right) \bmod m_{i_2};$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$c_{k_j} = A'_{i_j} \otimes b_{i_j} = \left(a'_{ij} \otimes b_{i_j}\right) \bmod m_{i_j}; \qquad (6)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$c_{k_n} = A'_{in} \otimes b_{in} = \left(a'_{i_n} \otimes b_{i_n}\right) \bmod m_{i_n}.$$

Логично, что число, полученное в (5) будет также ошибочным. В свою очередь в (6) происходило сложение не целых чисел, а только остатков по одному из соответствующих модулей.

Учитывая, что в (4) присутствовала верная цифра, а все остальные являлись ошибочными, то и после выполнения последующей операции в (6) будет одна верная цифра. Как уже говорилось, все основания в (6) соответствуют тем же основаниям в (5). Соответственно, если подставить из (6) в (5) верную цифру, то число $\tilde{A}$ будет исправлено и при проверке условие (1) будет выполняться. Если же подставить неверную цифру, то в числе $\tilde{A}$ будет уже двухкратная ошибка, которая при обнаружении укажет, что цифра a_{i_1} исправлена неверно и соответственно A'_{i_1} будет исключено из дальнейшего поиска ошибочной цифры. В случае если после проверки на выполнимость условия (1) окажется, что несколько цифр A'_{i_1} удовлетворяют ему - выполняется следующая по очереди арифметическая операция. Так продолжается до тех пор, пока не останется только одна скорректированная цифра, которая при замене цифры по соответствующему основанию в числе $\tilde{A}$ сделает его правильным. В этом случае данная цифра окончательно заменяет неправильную в числе $\tilde{A}$.

Таким образом, процедура реализующая предлагаемый метод параллельной коррекции заключается в исправлении цифр отдельно по всем основаниям, попавшим в первоначально определенную альтернативную совокупность искаженного числа $\tilde{A}$. После этого выполняются последующие запланированные в вычислительной цепи операции и проверяется - возвращает ли истинность числу $\tilde{A}$ каждая исправленная цифра в совокупности с цифрами по остальным основаниям этого числа. Когда остается только одна цифра, которая превращает число $\tilde{A}$ в правильное – она заменяет в нем ошибочную цифру.

Стоит обратить внимание, что в данном методе альтернативная совокупность определяется только один раз при определении наличия ошибки. В каждом последующем действии, вместо нее, проводится сверка с вхождением, полученного заменами цифр числа, в диапазон [0, M). Соответственно нет нужды на каждом этапе определять УАС.

Модель информационной технологии реализуемой подсистемой контроля и диагностики ошибок СПОИ, функционирующего в МСС, методом параллельной коррекции представлена на рис. 7.

Экспериментальная оценка разработанного метода параллельной коррекции показала, что в большинстве случаев локализация и исправление ошибки при I варианте вычислительной цепи происходит уже после второго действия. (рис. 8).

При II варианте вычислительной цепи локализация и исправление ошибки происходит не дольше чем за два действия, а в 40,6% случаев уже после первого действия (рис. 9).

При III варианте вычислительной цепи в 100% случаев локализация и коррекция ошибки происходит уже после первого действия (рис. 10).

Анализ полученных результатов показал, что разработанный метод параллельной коррекции одинаково эффективен при возникновении ошибки по любому из оснований; не нуждается в постоянном расчете и стягиванию ошибочных оснований, что присуще методу стягивания, способен выполнять контроль и диагностику в динамике вычислительного процесса и одинаково эффективен для различных вариантов вычислительной цепи.

Выводы

1. Проведенный анализ известных методов контроля и диагностики ошибок СПОИ функционирующего в МСС, к которым относятся метод стягивания альтернативных совокупностей и метод принятия гипотезы об ошибочном основании показал, что рассмотренные методы не способны одинаково эффективно решать задачи контроля и диагностики ошибок для различных вычислительных цепей, при различных вариантах распределения ошибки.

2. Предложенный в статье новый метод параллельной коррекции, в отличие от известных методов контроля и диагностики ошибки имеет следующие достоинства:

✓ предложенный метод не чувствителен к используемым в вычислительной цепи операциям и в большинстве случаев справляется с задачей контроля и диагностики ошибки за одно – два действия;

✓ разработанный метод способен эффективно производить контроль и диагностику ошибки, не зависимо от основания, в котором она возникла;

✓ благодаря способности определять и диагностировать ошибку за одно – два действия метод параллельной коррекции можно использовать как в коротких так и в длинных вычислительных цепях;

✓ метод лишен неоднозначности метода стягивания, в котором варианты возможных ошибочных оснований сводятся чаще всего к двум (т.к. контрольное всегда входит в АС);

✓ описанный метод не требует возврата к месту коррекции в случае ее ошибочности, как это происходит в методе принятия гипотезы. В случае определения ошибочности скорректированной цифры, она просто перестает обрабатываться в последующих действиях;

✓ метод позволяет производить контроль и диагностику ошибки, появляющийся в результате самокоррекции числа, в случаях, когда она превращает проверяемое число в правильное, но не истинное;

✓ для реализации данного метода не нужно выполнять алгоритмически сложных операций (как в случае с нулевизацией), что упрощает структуру использующего его СПОИ.

3. Полученные результаты позволяют сделать вывод о необходимости дальнейших, более системных исследований разработанного метода параллельной к оррекции, что позволит в последующем

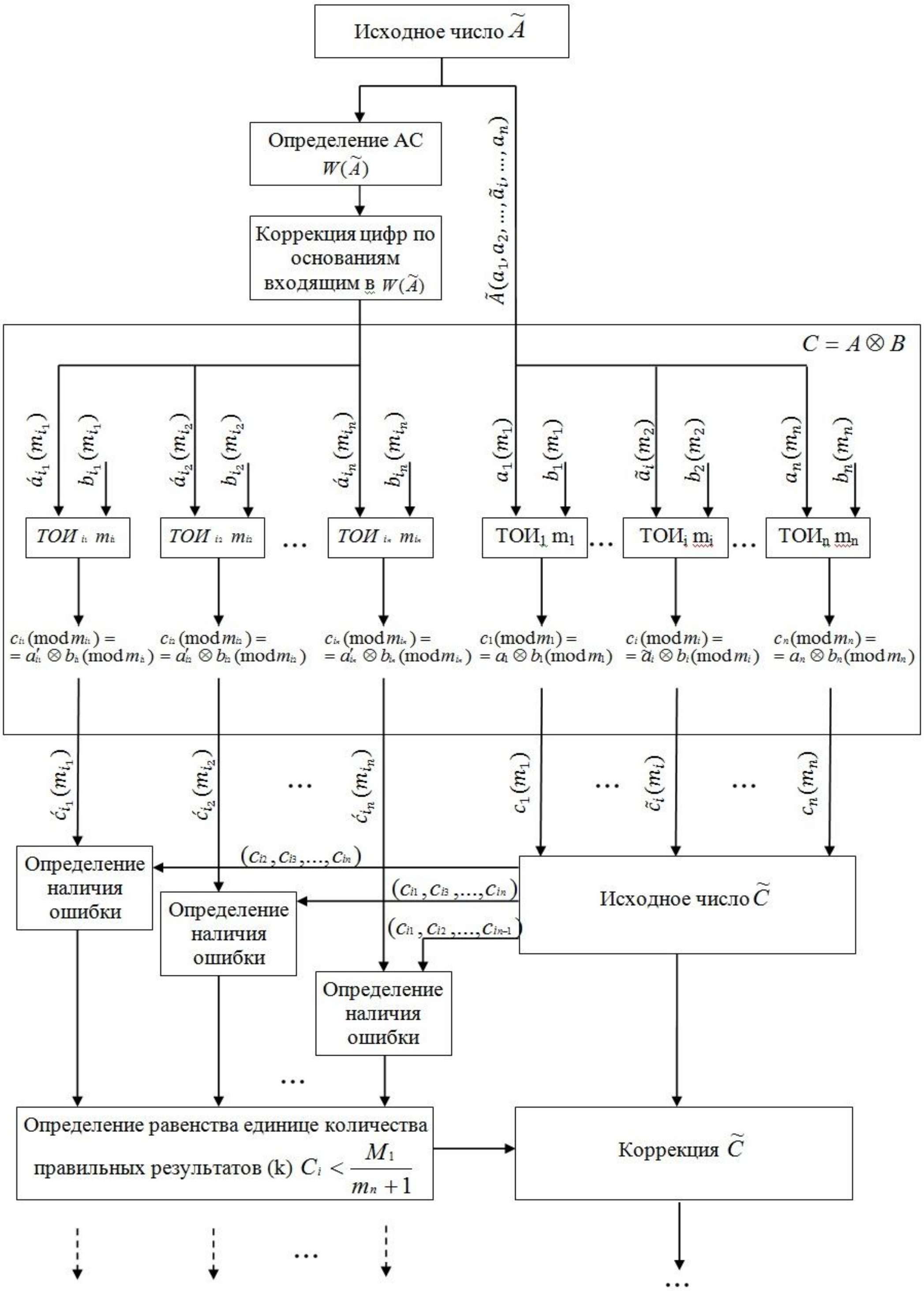

Рис. 7. Модель информационной технологии реализуемой подсистемой контроля и диагностики ошибок СПОИ, функционирующего в МСС, при реализации метода параллельной коррекции

внедрить его в качестве подсистемы контроля и диагностики ошибок спецпроцессора обработки информации функционирующего в МСС.

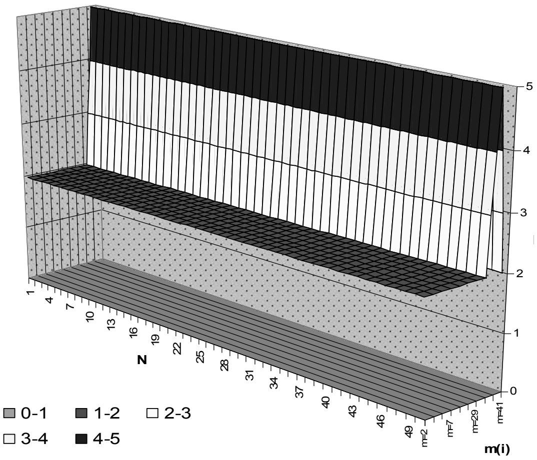

Рис. 8. Длина цепи операций до обнаружения места ошибки методом параллельной коррекции для I варианта вычислительно цепи

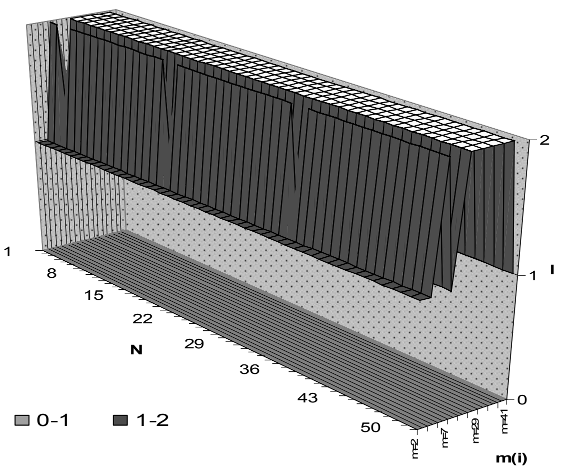

Рис. 9. Длина цепи операций до обнаружения места ошибки методом параллельной коррекции для II варианта вычислительно цепи

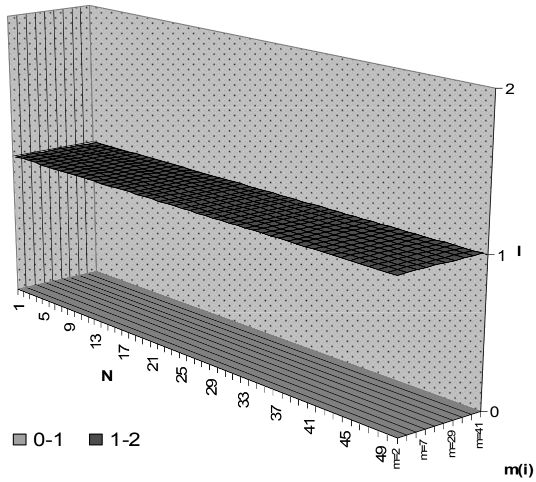

Рис. 10. Длина цепи операций до обнаружения места ошибки методом параллельной коррекции для III варианта вычислительно цепи

Список литературы

1. *Методы многоверсионной обработки информации в модулярной арифметике: моногр. / [В.И. Барсов, В.А. Краснобаев, А.А. Сиора, И.В. Авдеев]. – Х.: МОН, УИПА, 2008. – 460 с.*

2. *Модели и методы повышения отказоустойчивости и производительности управляющих вычислительных комплексов специализированных систем управления реального времени на основе применения непозиционных кодовых структур модулярной арифметики: моногр. / [В.И/ Барсов, Л.С/ Сорока, В.А. Краснобаев, Хери Али Абдуллах]. – Х.: МОН, УИПА, 2008. – 147 с.*

3. *Барсов В.И. Методология параллельной обработки информации в модулярной системе счисления: моногр. / В.И. Барсов, Л.С. Сорока, В.А. Краснобаев – Х.: МОН, УИПА, 2009.- 288 -с.*

4. *Модели и методы параллельной реализации логических операций в АСУ ТП: моногр. / [В.И. Барсов, В.А. Краснобаев, И.А. Фурман, и др.]. – Х.: МОНУ, УИПА, 2009. - 140 с.*

Поступила в редколлегию 15.04.2014

Рецензент: д-р техн. наук, проф. Г.А. Кучук, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

ДОСЛІДЖЕННЯ МЕТОДІВ КОНТРОЛЮ ТА ДІАГНОСТИКИ ПОМИЛОК СПЕЦПРОЦЕСОРА ОБРОБКИ ІНФОРМАЦІЇ, ЩО ФУНКЦІОНУЄ НА ОСНОВІ КОДІВ МОДУЛЯРНОЇ СИСТЕМИ ЧИСЛЕННЯ

В.І. Барсов, Є.О. Сотник, В.А. Краснобаев

Розглянуто та проаналізовано переваги та недоліки відомих методів контролю та діагностики помилок СПОІ, що функціонує в МСС. Розроблено та досліджено метод паралельної корекції, здатного підвищити ефективність контролю та діагностики помилок, незалежно від виду обчислювальної ланки і особливостей розподілу помилки в ній

***Ключові слова**: модулярна система числення, метод стягування альтернативних сукупностей, метод прийняття гіпотези про помилковість підстави, метод паралельної корекції, спецпроцесор обробки інформації*

RESEARCH METHODS FOR MONITORING AND DIAGNOSTIC INFORMATION PROCESSING ERROR OF SPECIAL PROCESSOR THAT OPERATES ON THE BASIS OF MODULAR CODE RADIX

V.I. Barsov, E.O. Sotnik, V.A. Krasnobaev

Reviewed and analyzed the advantages and disadvantages of the known methods of control and the diagnostics errors a specially designed information processing functioning in modular system value. Developed and studied a method of the parallel correcting that can increase the effectiveness of monitoring and fault diagnosis, regardless of the type of computer circuits and features of the distribution of the error in it

***Keywords**: modular system of calculation, method contraction alternative aggregates, the method of acceptance of the hypothesis of mistaken ground, parallel method of correction, a specially designed information processing*