



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 11/14 (2019.05)

(21)(22) Заявка: 2018109822, 20.03.2018

(24) Дата начала отсчета срока действия патента:
20.03.2018

Дата регистрации:
03.09.2019

Приоритет(ы):

(22) Дата подачи заявки: 20.03.2018

(45) Опубликовано: 03.09.2019 Бюл. № 25

Адрес для переписки:

214027, г. Смоленск, ул. Котовского, 2, ВА
ВПВО ВС РФ

(72) Автор(ы):

Константиновский Валентин Михайлович (RU),

Скачков Сергей Анатольевич (RU),

Ковалев Артем Александрович (RU),

Клюев Алексей Васильевич (RU),

Иванов Дмитрий Николаевич (RU)

(73) Патентообладатель(и):

Федеральное государственное казенное
военное образовательное учреждение
высшего образования "Военная академия
войсковой противовоздушной обороны
Вооруженных Сил Российской Федерации
имени Маршала Советского Союза А.М.
Василевского" Министерства обороны
Российской Федерации (RU)

(56) Список документов, цитированных в отчете
о поиске: US 2017/0323240 A1, 09.11.2017. WO
2017/074770 A1, 04.05.2017. US 2017/0262360 A1,
14.09.2017. RU 2637426 C1, 04.12.2017. RU
2455681 C1, 10.07.2012.

(54) Способ обеспечения надежности вычислительных систем при различных архитектурах избыточности

(57) Реферат:

Изобретение относится к вычислительной технике, в частности к системам, обладающим высоким уровнем отказоустойчивости при обработке информации, к которым предъявляются высокие требования по надежности как аппаратных, так и программных средств. Технический результат предлагаемого изобретения - повышение надежности вычислительных систем при различных архитектурах избыточности. В способе эффективность использования различных видов избыточности определяется по коэффициенту

прореживания потока сбоев $k_{пр}$, определение величины которого выполняется в следующей последовательности: построение двух моделей вычислительной системы с многовидовой избыточностью, имитация сбоев с помощью модуля генерации сбоев, с помощью модуля генератора тестовых воздействий формируют тестовые воздействия, статистически обрабатывают полученные данные на устройстве оценки сбоеустойчивости вычислительной системы. 8 ил.



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC
G06F 11/14 (2019.05)

(21)(22) Application: **2018109822, 20.03.2018**

(24) Effective date for property rights:
20.03.2018

Registration date:
03.09.2019

Priority:

(22) Date of filing: **20.03.2018**

(45) Date of publication: **03.09.2019** Bull. № 25

Mail address:

**214027, g. Smolensk, ul. Kotovskogo, 2, VA VPVO
VS RF**

(72) Inventor(s):

**Konstantinovskij Valentin Mikhajlovich (RU),
Skachkov Sergej Anatolevich (RU),
Kovalev Artem Aleksandrovich (RU),
Klyuev Aleksej Vasilevich (RU),
Ivanov Dmitrij Nikolaevich (RU)**

(73) Proprietor(s):

**Federalnoe gosudarstvennoe kazennoe voennoe
obrazovatelnoe uchrezhdenie vysshego
obrazovaniya "Voennaya akademiya vojskovoj
protivovozdushnoj oborony Vooruzhennykh Sil
Rossijskoj Federatsii imeni Marshala Sovetskogo
Soyuza A.M. Vasilevskogo" Ministerstva
oborony Rossijskoj Federatsii (RU)**

(54) **METHOD OF PROVIDING RELIABILITY OF COMPUTER SYSTEMS WITH DIFFERENT REDUNDANCY ARCHITECTURES**

(57) Abstract:

FIELD: computer equipment.

SUBSTANCE: invention relates to computer engineering, particularly to systems having high level of fault tolerance when processing information, which are high requirements for reliability of both hardware and software. In the method efficiency of using different types of redundancy is determined from the error flow reduction factor k_{pr} , determination of which value is performed in the following sequence: construction of

two models of computer system with multi-view redundancy, simulation of faults using fault generation module, using the test effects generator module generating test actions, statistically processing the obtained data on the computer system failure stability evaluation device.

EFFECT: high reliability of computer systems with different redundancy architectures.

1 cl, 8 dwg

R U 2 6 9 9 0 5 6 C 1

R U 2 6 9 9 0 5 6 C 1

Изобретение относится к вычислительной технике, в частности к системам обладающим высоким уровнем отказоустойчивости при обработке информации, к которым предъявляются высокие требования по надежности, как аппаратных, так и программных средств.

Известен ряд технических решений [1, 2, 3], позволяющих за счет введения различных видов избыточности в структуру вычислительной системы, обеспечить требуемый уровень надежности, однако такие решения не позволяют обеспечить оптимального распределения имеющихся ресурсов для получения требуемого уровня надежности.

Наиболее близким к предлагаемому техническому решению является подход предложенный в работе [4], в котором для оптимизации сложной системы обработки информации по критерию надежности определяется вариант системы и способы надежностной реализации каждой из подсистем, которые доставляют экстремум целевой функции P и обеспечивают успешное решение задач обработки информации с вероятностями не ниже заданных, в условиях ограниченности затрат на проект информационной системы. При этом процесс решения оптимизационной задачи представляет собой многоступенчатую структуру, каждая ступень которой связана с проверкой наличия тех или иных свойств у подмножества вариантов, что ведет к непосредственному сокращению исходного множества вариантов. Предлагаемый в работе [4] алгоритм использует методологию последовательного анализа и отсеивания вариантов путем отсеивания бесперспективных компонент кластерной структуры информационной системы как по ограничениям, так и по целевой функции без построения начальных частей вариантов и их дальнейшего развития.

Следуя [4] предлагается рассматривать систему обработки информации, которая состоит из отдельных подсистем, каждая из которых может быть реализована $u_{i(lj)}$ способами. Отказ любой из подсистем приводит к отказу всей системы в целом. Определение варианта системы осуществляется путем выбора способа реализации каждой подсистемы v_{extr} , доставляющей максимум целевой функции

$$P(v) = \prod_{j=1}^n P_j(u_{j(lj)})$$

при наличии ограничений

$$g_p(v) = \sum_{j=1}^n g_p(u_{j(lj)}) \leq g_p^*, \quad (p = 1, \dots, q),$$

$$g_p(v) = \sum_{j=1}^n g_p(u_{j(lj)}) \geq g_p^*, \quad (p = q + 1, \dots, Q),$$

$$v \in V, u_{j(lj)} \in U_j, \quad (j = 1, \dots, n),$$

где $U_j = \{u_{j(1)}, \dots, u_{j(lj)}, \dots, u_{j(\xi_j)}\}$ ($j=1, \dots, n$)-совокупность компонент различных типов, которые могут использоваться в j -й подсистеме (количество элементов в множестве U_j равно ξ_j), v - текущий вариант системы $V = \prod_{j=1}^n U_j$;

$P_j(u_{j(lj)})$ - надежность (вероятность безотказной работы на заданном интервале времени) элемента j -й подсистемы l_j -го типа;

$g_p(u_{j(lj)})$ - значение p -го ограничивающего фактора, для элемента l_j -го типа j -й подсистемы;

$g_p(v)$ - количество p -го ограничивающего фактора израсходованного на всю систему;

g_p^* - максимально возможное количество p -го ограничивающего фактора для всей системы в целом.

Представленная выше подстановка эквивалентна следующей задаче оптимизации

$$f(v) = \sum_{j=1}^n f_j(u_{j(lj)}) \rightarrow \max$$

при наличии указанных ограничений и с учетом, что $f_j(u_{j(lj)}) = \lg P_j(u_{j(lj)})$, так как логарифм является монотонным преобразованием. Для таких функций применяется условие алгоритма отсева по ограничениям [3], нахождение «безусловного» оптимального варианта для каждой рассматриваемой функции (как и проверка условия отсева) требует $\sum_{j=1}^n k_j$ вычислений значений этих функций не более

$1/2 \{ \sum_{j=1}^n k_j (k_j + 1) \}$ сравнений этих значений.

Для повышения надежности проектируемой системы обработки информации путем резервирования определяется максимальная $\lambda_{j(lj)}^*$ и минимальная $\lambda_{j(lj)}^{**}$ кратность резервирования (максимальное и минимальное возможное число резервных элементов) для l_j -го типа j -й подсистемы. Используются постоянные для каждого модуля величины

$$\Delta g_p^j = g_p^* - g_p(v_{v(l)} \setminus U_j)$$

($p=1, \dots, Q, j=1, \dots, n$).

Максимальную кратность резервирования $\lambda_{j(lj)}^*$ для каждого элемента определяется по формуле:

$$\lambda_{j(lj)}^* = \min_{p=1, \dots, Q} \left[\frac{\Delta g_p^j}{g(u_{j(lj)})} - 1 \right].$$

Такой способ позволяет определить вводимую избыточность резервных элементов и произвести оптимальный выбор состава компонентов, используемых при формировании надежных структур систем обработки информации, реализуемых в дальнейшем использовании с использованием кластеризации.

Недостатком рассмотренного способа [4] является то, что в нем повышение надежности производится только путем введения структурной избыточности и не рассматриваются вопросы комплексной оптимизации различных видов избыточности. Это не позволяет обеспечить эффективное распределение ресурсов для обеспечения высокого уровня надежности вычислительной системы по отношению как к устойчивым отказам, так и сбоям, наиболее характерным видам отказов для цифровой техники.

Технический результат предполагаемого изобретения - повышение надежности вычислительных систем при различных архитектурах избыточности.

Технический результат достигается тем, что в известном способе [4] предлагается рассматривать вычислительную систему, которая состоит из отдельных подсистем,

каждая из которых может быть реализована способами. Отказ любой из подсистем приводит к отказу всей системы в целом. Определение варианта системы осуществляется путем выбора способа реализации каждой подсистемы v_{extr} , доставляющий максимум целевой функции

$$P(v) = \prod_{j=1}^n P_j(u_{j(lj)})$$

при наличии ограничений

$$g_p(v) = \sum_{j=1}^n g_p(u_{j(lj)}) \leq g_p^*, \quad (p = 1, \dots, q),$$

$$g_p(v) = \sum_{j=1}^n g_p(u_{j(lj)}) \geq g_p^*, \quad (p = q + 1, \dots, Q),$$

$$v \in V, u_{j(lj)} \in U_j, \quad (j = 1, \dots, n),$$

где $U_j = \{u_{j(1)}, \dots, u_{j(lj)}, \dots, u_{j(\xi_j)}\}$ ($j=1, \dots, n$) - совокупность компонент различных типов, которые могут использоваться в j -й подсистеме (количество элементов в

множестве U_j равно ξ_j), v - текущий вариант системы $V = \prod_{j=1}^n U_j$;

$P_j(u_{j(lj)})$ - надежность (вероятность безотказной работы на заданном интервале времени) элемента j -й подсистемы l_j -го типа;

$g_p(u_{j(lj)})$ - значение p -го ограничивающего фактора, для элемента l_j -го типа j -й подсистемы;

$g_p(v)$ - количество p -го ограничивающего фактора израсходованного на всю систему;

g_p^* - максимально возможное количество p -го ограничивающего фактора для всей системы в целом.

Представленная выше подстановка эквивалентна следующей задаче оптимизации

$$f(v) = \sum_{j=1}^n f_j(u_{j(lj)}) \rightarrow \max$$

при наличии указанных ограничений и с учетом, что $f_j(u_{j(lj)}) = \lg P_j(u_{j(lj)})$, так как логарифм является монотонным преобразованием. Для таких функций применяется условие алгоритма отсева по ограничениям [3], нахождение «безусловного» оптимального варианта для каждой рассматриваемой функции (как и проверка условия отсева) требует $\sum_{j=1}^n k_j$ вычислений значений этих функций не более

$1/2 \{ \sum_{j=1}^n k_j (k_j + 1) \}$ сравнений этих значений. Сначала выполняется отсев элементов

по типам ограничений без применения резервирования, в ходе которого множество возможных реализаций вычислительной системы V преобразуется в множество

$V^* = \prod_{j=1}^n U_j^*$, где U_j^* - совокупность компонент различных типов, которые могут

использоваться в j-й подсистеме после отсева по ограничениям.

Для повышения надежности проектируемой вычислительной системы разрабатывается вариант совместного использования различных видов избыточности (структурной, временной, информационной, алгоритмической). Возможность их группирования достигается за счет комплексирования теории алгебры логики, с помощью которой моделируются собственно алгоритм функционирования, а также структурная избыточность вычислительной системы и программного обеспечения (математического пакета), позволяющего имитировать другие виды избыточности.

Эффективность использования различных видов избыточности определяется по коэффициенту прореживания потока сбоев $k_{пр}$, определение величины которого выполняется в следующей последовательности:

- 1) построение двух моделей вычислительной системы с многовидовой избыточностью;
- 2) имитация сбоев;
- 3) создание генератора тестовых воздействий;
- 4) оценка показателя сбоеустойчивости вычислительной системы.

Процедура построения моделей вычислительной системы включает: получение сведений о номенклатуре, серии и количестве ИМС, поиск в базе данных или построение их булевых моделей, выполнение их функционального связывания по принципиальной схеме; создание и реализацию в математическом пакете алгоритмов (фиг. 1, фиг. 2, фиг. 3, фиг. 4). Создание алгоритмов структурной (фиг. 1), временной (фиг. 2), информационной (фиг. 3) и алгоритмической (фиг. 4) избыточности предполагает корректировку моделей вычислительной системы, соответственно множество возможных реализаций вычислительной системы V^* преобразуется в множество $V^{**} = \prod_{j=1}^n U_j^{**}$,

где U_j^{**} - совокупность компонент различных типов, которые могут использоваться в j-й подсистеме после введения избыточности с учетом возможных допусков для каждого из ограничивающих факторов.

Имитация сбоев в модели вычислительной системы предполагает: определение места, времени и кратности (закона распределения плотности вероятности потока сбоев) на основе априори известных статистических данных по сбоям, например по критерию Колмогорова или χ^2 ; синтез алгоритма внесения сбоя кратностью ν в модель, т.е. выбор из множества переменных n модели $\{x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_{i+1+m}, x_{i+1+m+1}, \dots, x_n\}$ ν элементов, замена для каждого выбранного элемента логического состояния на противоположное.

Создание генератора тестовых воздействий. Тестовое воздействие представляет собой вектор, состоящий из последовательности логических «0» и логических «1» длиной

$L_T = \sum_{i=1}^k L_{Ti}$, где L_{Ti} - количество выходных переменных i-го абонента, k - число

абонентов вычислительной системы.

Метод формирования тестового воздействия (последовательного перебора, случайный, детерминированных направлений) выбирается при настройке генератора.

Перечисленные методы манипулируют множеством (фиг. 5), размерностью, равной $L_{Ti}=2^n$, где n - разрядность шины данных. Метод детерминированных направлений реализуется режимом фильтрации. Применением фильтрации достигается сокращение величины L_{Ti} , а значит, и временных затрат на вычисление показателя сбоеустойчивости.

Правило работы фильтра

$F: x_1, x_2, \dots, x_i, \dots, x_n \in M_x | x_1, x_2, \dots, x_i, \dots, x_n \sim y_1, y_2, \dots, y_i, \dots, y_m.$

В результате фильтрации показатель мощности множества входных кодовых наборов $P(M_x)$ уменьшается, как минимум, на величину $\Delta P(M_x) > 2^n - 2^m$, если $n > m$. Для случая

$n \leq m$ величина $\Delta P(M_x)$ определяется статистическим перебором множества $P(M_x) = 2^n$ кортежей входных кодовых наборов и выбором из этого множества кортежей, приводящих к изменению выходных кортежей множества $P(M_y) = 2^m$ (на фиг. 5 отфильтрованные множества входных и выходных кортежей выделены серым тоном).

Оценивать $k_{пр}$ предлагается в соответствии с информационно-временной статистической моделью изображенной на фиг. 6.

Устройство управления - 1 (фиг. 6) координирует работу модулей модели согласно заданному алгоритму (фиг. 7, фиг. 8) и реализуется в математическом пакете программным способом.

Модуль генератора сбоев - 2 (фиг. 6) вносит по выбранному закону сбой в модель вычислительной системы №2 - 5 (фиг. 6), с определенной интенсивностью,

характеризующейся значением $\Delta n_c / t$ - количеством сбоев внесенных в модель вычислительной системы единицу времени.

Модуль генератора тестовых воздействий - 3 (фиг. 6) формирует тестовые воздействия с использованием рассмотренного выше метода и подает их на модели вычислительных систем №1 и 2 - (4 и 5 на фиг. 6).

Модель вычислительной системы №1 - 4 (фиг. 6) формирует массив эталонных откликов $M_{эт}$ на тестовые воздействия.

Модель вычислительной системы №2 - 5 (фиг. 6) формирует массив откликов M_c в условиях имитации сбойных ситуаций.

Схема сравнения - 6 (фиг. 6) проверяет соответствие откликов с выходов моделей вычислительных систем (4 и 5 на фиг. 6), и определяет $\Delta n_{осх} / t$ - количество отказов

сбойного характера (сбоев не перехваченных моделью вычислительной системы с конкретной архитектурой избыточности) в единицу времени по количеству несовпадений $M_{эт}$ и M_c .

Модуль оценки сбоеустойчивости - 7 (фиг. 6) определяет величины показателей сбоеустойчивости по результатам обработки данных, полученных в ходе статистического исследования моделей вычислительных систем (4 и 5 на фиг. 6) по формуле

$$k_{пр} = \frac{\Delta n_{осх} / t}{\Delta n_c / t} = \frac{\lambda_{осх}}{\lambda_c},$$

где $\lambda_{осх}$ - интенсивность потока отказов сбойного характера на выходе модели вычислительной системы, λ_c - интенсивность потока сбоев вносимых в модель вычислительной системы.

Из отобранных вариантов реализации вычислительной системы V^{**} выбирается вариант обладающий максимальным значением $k_{пр}$.

Информационно-временная статистическая модель (фиг. 6) позволяет решать задачу количественной оценки сбоеустойчивости в условиях различных сочетаемости и перекрытия видов избыточности, оценивать эффективность системы обнаружения сбоев, определять коэффициент функциональной загрузки подсистем

вычислительной системы.

Источники информации

1. Научный журнал «Математические машины и системы» 2002 г. №2 Теслер Г.С. «Концепция создания вычислительных средств с высоким уровнем отказоустойчивости».

2. Патент на изобретение «Способ автоматического управления избыточностью неоднородной вычислительной системы и устройство для его реализации» RU 2612569.

3. Вычислительные методы исследования и проектирования сложных систем.

Михалевич В.С., Волкович В.Л. - М.: Наука. Главная редакция физико-математической литературы, 1982.

4. Электронный журнал «ИССЛЕДОВАНО В РОССИИ» 1123 <http://zhurnal.ape.relarn.ru/articles/2004103.pdf>.

(57) Формула изобретения

Способ обеспечения надежности вычислительных систем при различных архитектурах избыточности, заключающийся в том, что при проектировании вычислительной системы ее разделяют на ряд подсистем, каждая из которых может быть реализована различными $u_{i(lj)}$ способами, определение варианта системы осуществляется путем выбора способа реализации каждой подсистемы v_{extr} , доставляющей максимум целевой функции

$P(v) = \prod_{j=1}^n P_j(u_{j(lj)})$, при наличии ограничений $g_p(v) = \sum_{j=1}^n g_p(u_{j(lj)}) \leq g_p^*$, $(p=1, \dots, q)$, $g_p(v) = \sum_{j=1}^n g_p(u_{j(lj)}) \geq g_p^*$, $p=q+1, \dots, Q$, $v \in V$, $u_{j(lj)} \in U_j$, $(j=1, \dots, n)$,

где $U_j = \{u_{j(1)}, \dots, u_{j(lj)}, \dots, u_{j(\xi_j)}\}$, $(j=1, \dots, n)$ - совокупность компонент различных типов, которые могут использоваться в j -й подсистеме (количество элементов в множестве U_j равно ξ_j), v - текущий вариант системы $V = \prod_{j=1}^n U_j$, $P_j(u_{j(lj)})$ -

надежность (вероятность безотказной работы на заданном интервале времени) элемента j -й подсистемы l_j -го типа, $g_p(u_{j(lj)})$ - значение p -го ограничивающего фактора, для

элемента l_j -го типа j -й подсистемы, $g_p(v)$ - количество p -го ограничивающего фактора, израсходованного на всю систему, g_p^* - максимально возможное количество p -го ограничивающего фактора для всей системы в целом, при наличии указанных ограничений и с учетом, что $f_j(u_{j(lj)}) = \lg P_j(u_{j(lj)})$, находится максимум функции

$f(v) = \sum_{j=1}^n f_j(u_{j(lj)})$, сначала выполняется отсев элементов по типам ограничений без применения резервирования, в ходе которого множество возможных реализаций вычислительной системы V преобразуется в множество $V^* = \prod_{j=1}^n U_j^*$, где U_j^* -

совокупность компонент различных типов, которые могут использоваться в j -й подсистеме после отсева по ограничениям, после чего рассматриваются варианты возможного повышения надежности путем введения избыточности с учетом возможных допусков для каждого из ограничивающих факторов $\Delta g_p^j = g_p^* - g_p(v)$, отличающийся

тем, что для повышения надежности проектируемой вычислительной системы выбирается вариант совместного использования различных видов избыточности (структурной, временной, алгоритмической, информационной), эффективность варианта совместного использования различных видов избыточности определяется по коэффициенту прореживания потока сбоев $k_{пр}$, определение величины которого

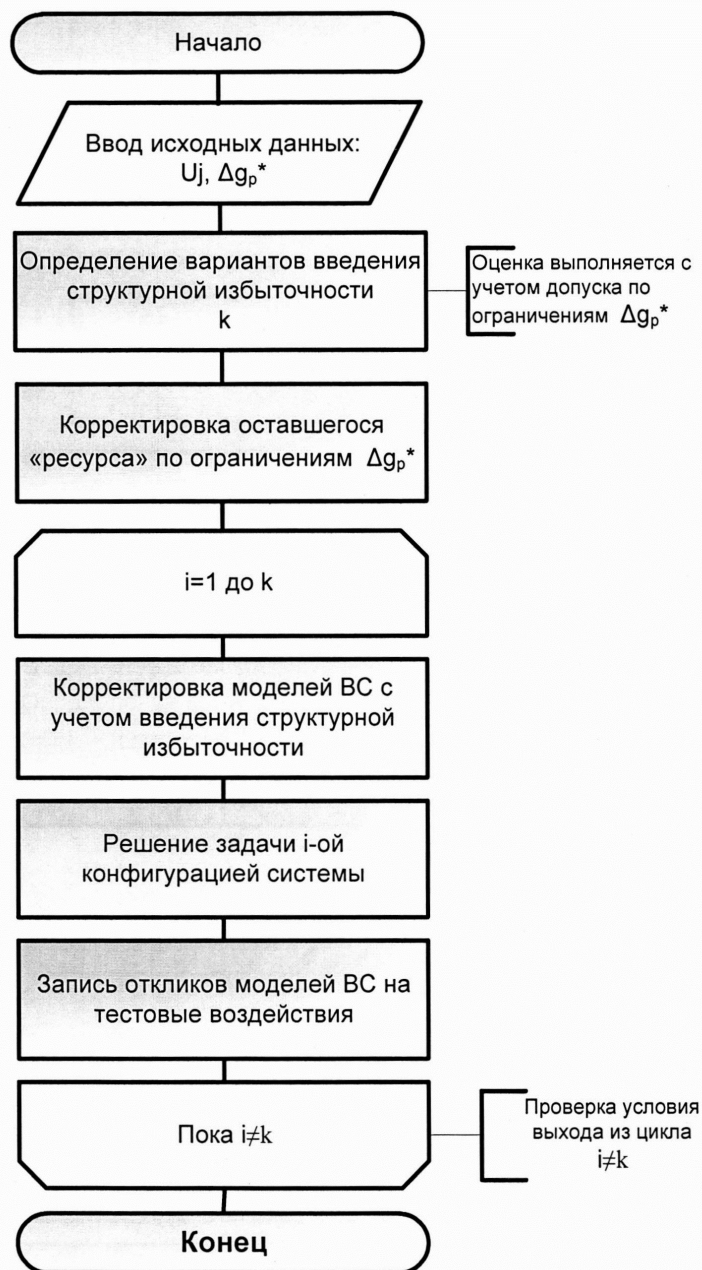
выполняется в следующей последовательности, последовательно на основе алгебры логики, формируются две идентичные модели каждого из отобранных вариантов построения вычислительной системы с возможными вариантами введения многовидовой избыточности, соответственно множество возможных реализаций вычислительной

системы V^* преобразуется в множество $V^{**} = \prod_{j=1}^n U_j^{**}$, где U_j^{**} - совокупность компонент различных типов, которые могут использоваться в j-й подсистеме после введения избыточности с учетом возможных допусков для каждого из ограничивающих факторов, выполняется имитация сбоев путем определения места, времени и кратности (закон распределения плотности вероятности потока сбоев) на основе априори известных статистических данных по сбоям, симитированные сбои вносятся в одну из моделей, с помощью модуля генератора сбоев, на основе одного из известных методов (например, метода последовательного перебора, метода детерминированных направлений, случайным способом), с помощью модуля генератора тестовых воздействий, формируются тестовые воздействия, которые подаются на обе модели вычислительной системы, выполняется оценка показателя сбоеустойчивости - $k_{пр}$ - варианта вычислительной системы путем сравнения откликов на тестовые воздействия с выходов обеих моделей вычислительной системы при помощи схемы сравнения и статистической обработки полученных результатов на устройстве оценки сбоеустойчивости, при этом расчет коэффициента прореживания сбоев выполняется по формуле

$$k_{пр} = \frac{\Delta n_{осх}/t}{\Delta n_c/t} = \frac{\lambda_{осх}}{\lambda_c},$$

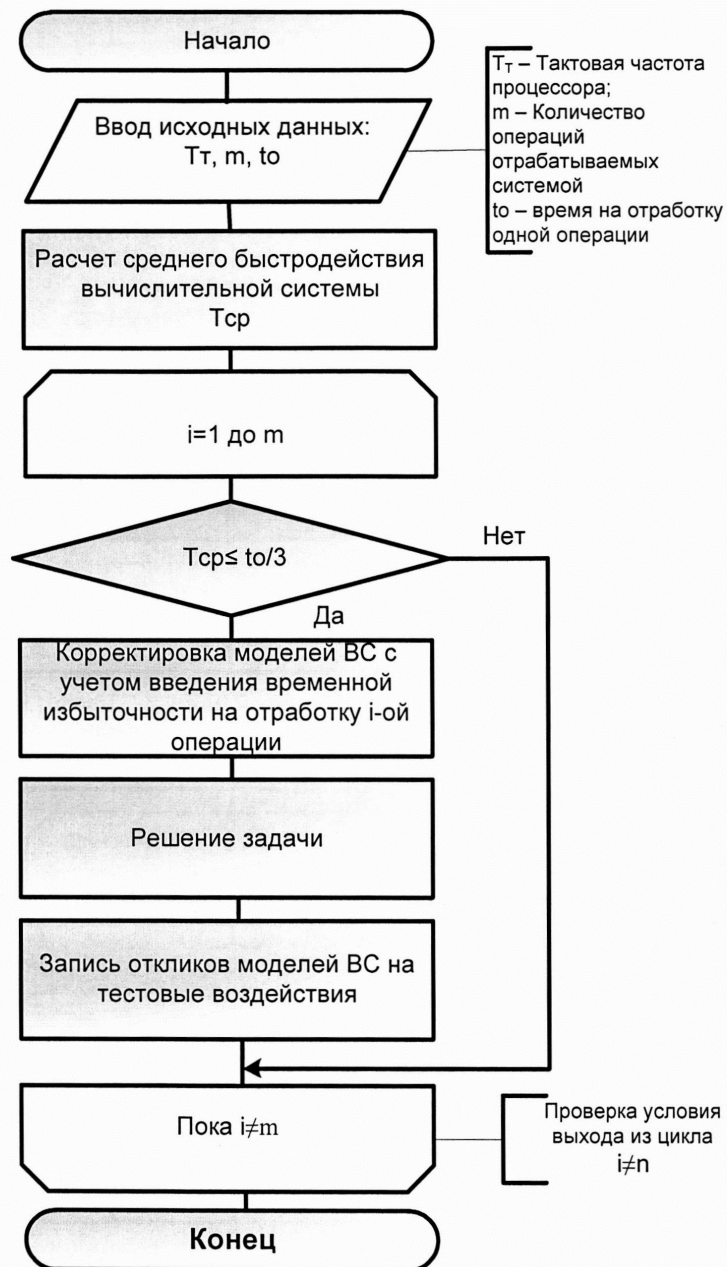
где $\Delta n_c/t$ - количество сбоев, внесенных в модель вычислительной системы в единицу времени, $\Delta n_{осх}/t$ - количество отказов сбойного характера (сбоев, не перехваченных моделью вычислительной системы с конкретной архитектурой избыточности) в единицу времени, $\lambda_{осх}$ - интенсивность потока отказов сбойного характера на выходе модели вычислительной системы, λ_c - интенсивность потока сбоев, вносимых в модель вычислительной системы, из отобранных вариантов реализации вычислительной системы V^{**} , выбирается вариант, обладающий максимальным значением $k_{пр}$.

1

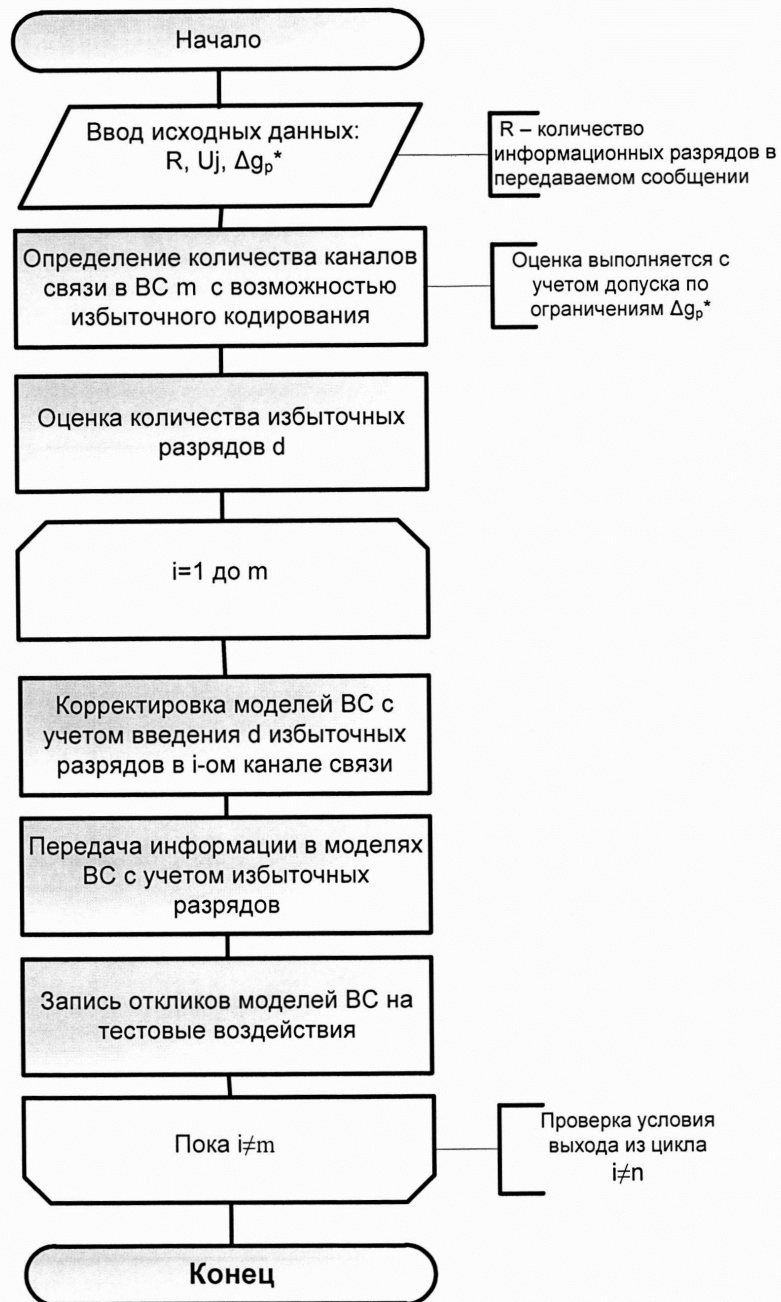


Фиг. 1

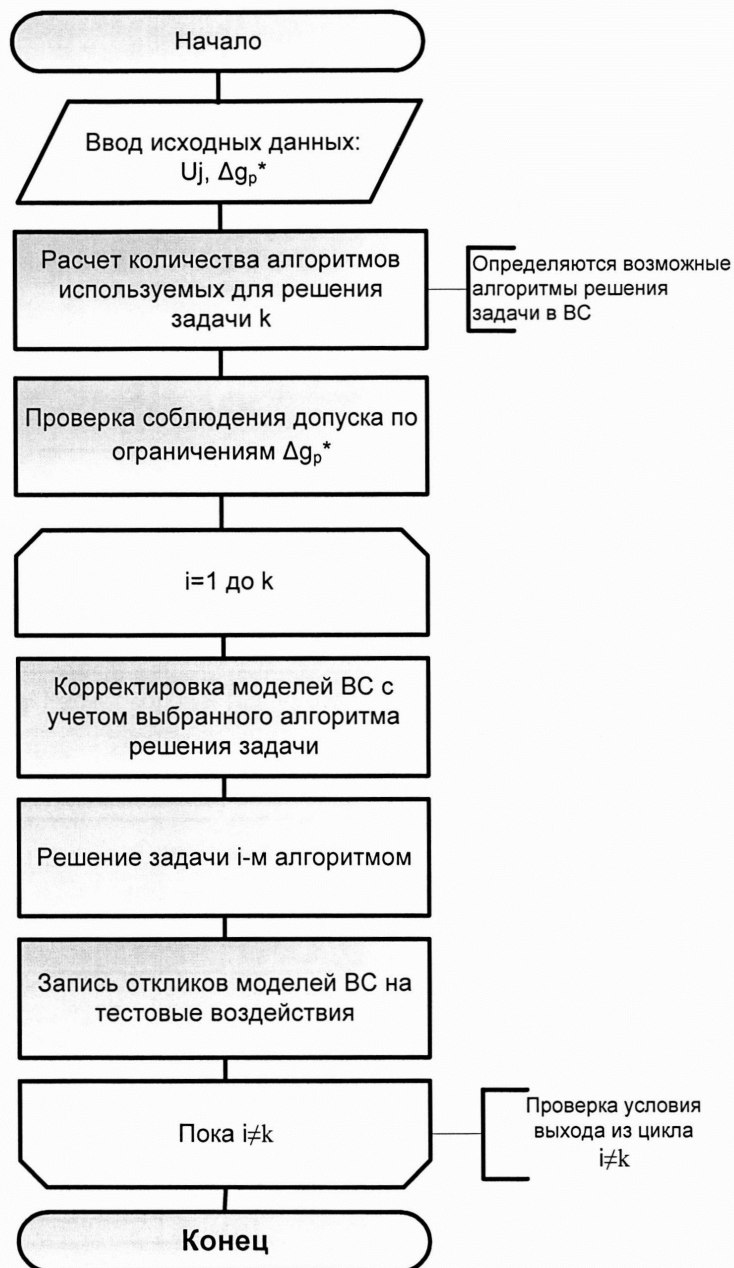
2



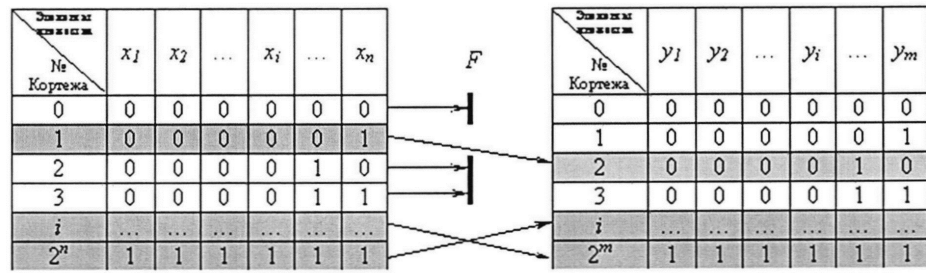
Фиг. 2



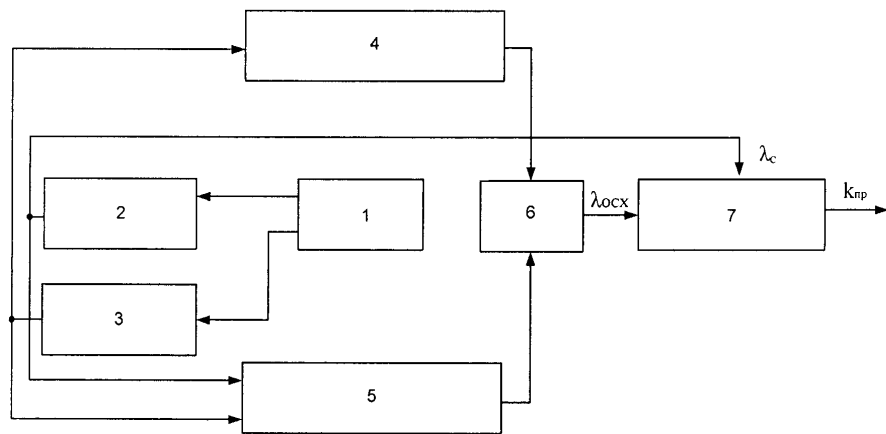
Фиг. 3



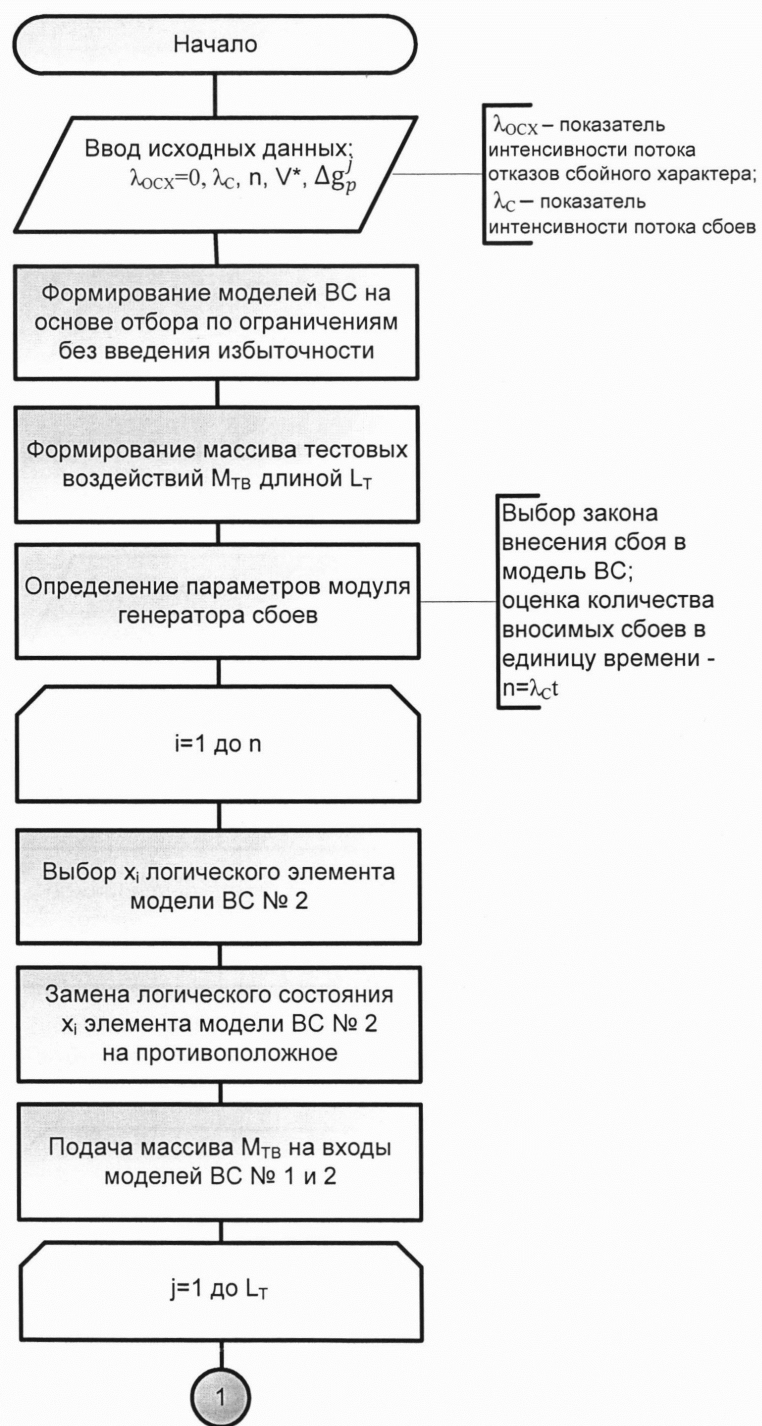
Фиг. 4



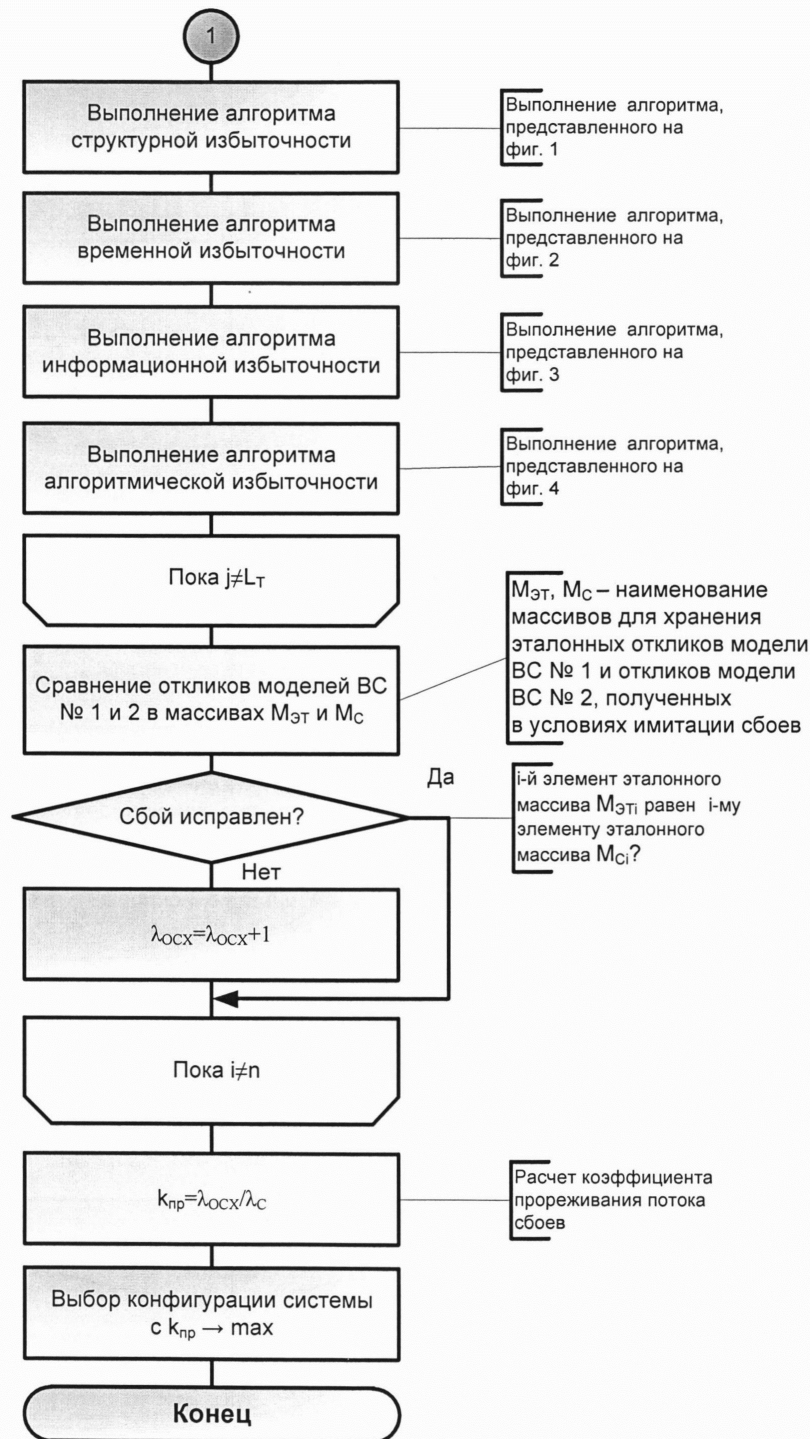
Фиг. 5



Фиг. 6



Фиг. 7



Фиг. 8