

Главным условием интеграции здесь выступают правила межсетевых обмена информацией [6]. Это дает возможность решения задач с использованием многомерного анализа данных любой территории проживания граждан.

Повышение эффективности процесса реги-

Список литературы

1. Глушков В.М. Основы безбумажной информатики [Текст] / В.М. Глушков; изд-е 2-е, испр. – М.: Наука, 1987. – 552 с.
2. Типовые требования к автоматизированным системам электронного документооборота. Спецификация MoReq [Электронный ресурс]. – Режим доступа: <http://www.cornwell.co.uk/moreq.html>
3. Методология структурного анализа и проектирования SADT [Электронный ресурс]. – Режим доступа: <http://vernikov.ru/krisis/item/210-sadt-metodology-structurnogoprojectirovanija.html>
4. Куликов Г.Г. Создание BPMN-моделей в программном продукте RunaWFE: лабораторный практикум [Текст] / Г.Г. Куликов, А.Г. Михеев, М.В. Орлов, Р.К. Габбасов, Д.В. Антонов. – Уфимск. гос. авиац. техн. ун-т, 2012. – 61с.
5. Информационно-аналитическая система на основе знаний регистрационного учета населения / Г. Г. Куликов, В. В. Антонов, А. А. Савина // Вестник УГАТУ. – 2008. – Т. 10. – № 2(27). – С. 60– 67.
6. Материалы федеральной целевой программы «Электронная Россия». Постановление №403 от 9.06.2010 г. «О внесении изменений в ФЦП «Электронная Россия» [Электронный ресурс]. – Режим доступа: <http://www.elrussia.ru>.

страционного учета гражданами органами государственной власти обеспечивает качественный эффект, связанный с разработкой системы, которая обеспечивает упрощение функций государственных услуг.

References

1. Glushkov V.M. Osnovy bezbumazhnoj informatiki [Tekst] / V.M. Glushkov; izd-e 2-e, ispr. – M.: Nauka, 1987. – 552 p.
2. Tipovye trebovanija k avtomatizirovannym sistemam jelektronного dokumentooborota. Specifikacija MoReq [Jelektronnyj resurs]. – Rezhim dostupa: <http://www.cornwell.co.uk/moreq.html>
3. Metodologija strukturnogo analiza i proektirovanija SADT [Jelektronnyj resurs]. – Rezhim dostupa: <http://vernikov.ru/krisis/item/210-sadt-metodology-structurnogoprojectirovanija.html>
4. Kulikov G.G. Sozdanie BPMN-modelej v programmnom produkte RunaWFE: laboratornyj praktikum [Tekst] / G.G. Kulikov, A.G. Miheev, M.V. Orlov, R.K. Gabbasov, D.V. Antonov. – Ufimsk. gos. aviac. tehn. un-t, 2012. – 61s.
5. Informacionno-analiticheskaja sistema na osnove znanij registracionnogo ucheta naselenija / G. G. Kulikov, V. V. Antonov, A. A. Savina // Vestnik UGATU. – 2008. – T. 10. – № 2(27). – P. 60– 67.
6. Materialy federal'noj celevoj programmy «Jelektronnaja Rossija». Postanovlenie №403 ot 9.06.2010 g. «O vnesenii izmenenij v FCP Jelektronnaja Rossija» [Jelektronnyj resurs]. – Rezhim dostupa: <http://www.elrussia.ru>.



Дегтярев А.Р.
Degtyarev A.R.
аспирант

кафедра «Измерительно-вычислительные комплексы», ФГБОУ ВО «Ульяновский государственный технический университет», Россия, г. Ульяновск



Киселев С.К.
Kiselev S.K.

доктор технических наук,
профессор, заведующий кафедрой
«Измерительно-вычислительные комплексы»,
ФГБОУ ВО «Ульяновский государственный
технический университет»,
Россия, г. Ульяновск

УДК 629.7.05

ОТКАЗОУСТОЙЧИВЫЕ РЕКОНФИГУРИРУЮЩИЕСЯ КОМПЛЕКСЫ ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИОНИКИ

В статье рассмотрены проблемы создания отказоустойчивых многопроцессорных комплексов бортового оборудования (КБО) на основе интегрированной модульной авионики (ИМА). Показано, что для построения перспективных систем ИМА требуется разработка комплексных алгоритмов их реконфигурации.

Определены три типа динамической реконфигурации и сформулированы соответствующие задачи. Предложены алгоритмы реконфигурации, включающие в себя распределение функциональных задач между доступными аппаратными ресурсами многопроцессорных комплексов ИМА по заданному критерию оптимальности, которые учитывают критичность выполняемых функций, количество исправных и отказавших модулей крейта, а также режим полета. Рассмотрены особенности отдельных видов реконфигурации, влияние на ее выполнение структуры многопроцессорного комплекса и типа применяемой операционной системы реального времени. Работа алгоритмов продемонстрирована на примере КБО вертолета, который способен функционировать в режиме крейсерского полета и режиме боя. Построены графы его функциональной и аппаратной структур, учитывающие критичность функций, потоки данных между ними и особенности аппаратной платформы, а также упорядоченные множества функций, которые соответствуют выбранному режиму функционирования комплекса. Показаны распределения функциональных задач КБО, полученные на основе предложенных алгоритмов. Рассмотрены особенности практической реализации реконфигурации, выявленные при обработке алгоритмов на макете мультипроцессорной системы.

Проведена оценка эффективности разработанных алгоритмов реконфигурации путем анализа деревьев неисправности для классической федеративной архитектуры с двойным резервированием и архитектуры ИМА, состоящей из одного крейта с резервированным коммутатором.

Даны рекомендации по усовершенствованию предложенных алгоритмов с целью осуществления частичной реконфигурации, не затрагивающей исправные модули, а так же обозначена невозможность реализации реконфигурации на существующих комплексах с динамически перестраиваемой архитектурой, не учитывающей особенностей построения систем ИМА. Предложенные алгоритмы могут использоваться для построения и исследования крейтов модульной авионики, а также для разработки математических моделей взаимодействия аппаратной и программной частей комплекса ИМА и их отказов.

Ключевые слова: многопроцессорные вычислительные системы, реконфигурация, алгоритм, интегрированная модульная авионика, дерево отказов, надежность, операционная система реального времени, крейт, распределение задач, оптимальность, граф.

FAULT-TOLERANT RECONFIGURABLE COMPLEXES OF INTEGRATED MODULAR AVIONICS

This paper surveys problems of development of fault-tolerant multiprocessor airborne complexes, which are based on Integrated Modular Avionics (IMA). It's shown that perspective reliable IMA system development is impossible without proper reconfiguration algorithms.

Three types of reconfiguration are determined and appropriate tasks are posed. Reconfiguration algorithms, which include functional tasks distribution among available IMA multiprocessor hardware resources, are presented in order to provide requisite optimal functionality. They are also adjusted for chosen optimal criterion, which includes functions criticality, amount of operational and invalid crate modules and the flight mode. Different features of all particular reconfiguration types are examined along with influence of multiprocessor system architecture and real-time operating system type on reconfiguration's implementation. Attack helicopter hardware complex, which is capable to function in cruise flight mode and combat mode, was considered as an example. Respective hardware and functional graphs were built, which are adjusted for functions criticality, data streams between them, hardware features and also streamlined set of functions that fit in given functionality mode of the complex. Given functional tasks of the complex are distributed with presented algorithms. Features of practical reconfiguration's implementation, which were revealed during the tests of algorithms on the multiprocessor system prototype, are also shown.

Evaluation of developed reconfiguration algorithms effectiveness was implemented with Fault Tree Analysis (FTA) for traditional federative architecture with double reservation and for IMA architecture that consists of one crate with reserved commutation module.

Recommendations are given in order to improve presented algorithms for performing partial reconfiguration which doesn't concern operational modules. It is shown that impossible to implement reconfigurable architecture based on existing dynamically reconstructive architecture which doesn't take in account IMA system features. Developed algorithms can be used to build and research IMA crates, its failures and for IMA hardware-to-software interactions mathematical models development.

Keywords: multiprocessor system, reconfiguration, algorithm, Integrated Modular Avionics, fault tree, reliability, real-time operating system, crate, tasks distribution, optimality, graph.

1. Постановка задач динамической реконфигурации

Динамическое изменение структуры комплекса бортового оборудования (КБО), спроектированного на основе концепции интегрированной модульной авионики (ИМА) и заключающееся в перераспределении программных и аппаратных ресурсов, на сегодняшний день является перспективным направлением развития авиационных систем, ведущим к созданию комплексов с повышенной надежностью и эффективностью выполнения задач [1].

Ключевой особенностью функционирования КБО ИМА является наличие программных приложений разного уровня критичности, которые выполняют заданные функции и разделяют общие аппаратные ресурсы. При этом массивы данных, которые являются результатами выполнения приложений, передаются по внутренней сети между модулями, устройствами ввода/вывода, датчиками и другими элементами управления. На основе информации

о структуре и взаимодействиях комплекса ИМА можно выделить три базовых категории отказов [2]:

- сбой в работе программного приложения или аппаратного модуля;
- неадекватная реакция системы на запрос от исправного приложения или модуля, то есть неверный результат распознавания инструкции комплексом или сбой при попытке изолирования некорректно функционирующего приложения;
- выполнение какой-либо функции при отсутствии запроса, то есть отказ, произошедший на уровне всей платформы и не относящийся только лишь к приложению или устройству.

На основании перечисленных отказных состояний были определены три вида динамической реконфигурации [3]:

- аппаратная динамическая реконфигурация. Осуществляется с целью перераспределения корректно функционирующих приложений по

исправным вычислительным модулям при аппаратном сбое;

- программная динамическая реконфигурация. Осуществляется с целью перераспределения корректно функционирующих приложений по исправным модулям с учетом выбранного критерия оптимальности в случае программного сбоя или при поступлении недостоверной информации от внешних устройств, которую нельзя использовать для вычислений;

- функциональная динамическая реконфигурация. Комбинированный вид реконфигурации, который осуществляется при смене выполняемой задачи, когда одна информация (например, данные от обзорной системы, системы управления вооружением и т.д.) становится более приоритетной, чем другая (например, навигация, автопилот и т.д.).

Для постановки задач реконфигурации необходимо представить КБО в виде какой-либо модели,

которая отражает его основные характеристики. Положим, что комплекс состоит из множества V типовых ВМ, расположенных внутри крейта. Пусть для реализации всех необходимых функций КБО требуется выполнять множество P приложений, работа которых контролируется операционной системой. В модулях также должна быть предусмотрена возможность размещения S массивов, которые являются исходными данными и результатами функционирования приложений [4].

Заданы размеры σ_s массивов $s \in S$ и приложений σ_p , $p \in P$, которые так же учитывают объем памяти, необходимый для хранения промежуточной информации. Заданы также частота v_p обращения приложения p к данным и объем вычислений ρ_p , необходимый для функционирования программ.

Для решения задачи размещения приложений и массивов данных в комплексе определим значение следующих булевых переменных:

$$x_{pv} = \begin{cases} 1, & \text{если массив } s \in S \text{ находится в памяти модуля } v \in V, \\ 0, & \text{если массив } s \in S \text{ не находится в памяти модуля } v \in V, \end{cases}$$

$$x_{pv} = \begin{cases} 1, & \text{если приложение } p \in P \text{ выполняется на модуле } v \in V, \\ 0, & \text{если приложение } p \in P \text{ не выполняется на модуле } v \in V. \end{cases}$$

При постановке задачи размещения нужно учитывать тип сети передачи данных. Внутрикрейтовые связи, как правило, основаны на каком-либо высокоскоростном интерфейсе. Например, PCI-Express предоставляет соединение типа точка-точка посредством коммутатора. Так как скорость обмена между модулями по такому каналу связи порядка

нескольких Гбит/с, то можно считать, что они соединены друг с другом парными связями по типу «каждый с каждым».

Загрузка λ_v памяти модуля $v \in V$ определяется размерами массивов данных и исходного кода приложения, размещенного в этом модуле:

$$\lambda_v = \sum_{s \in S} \sigma_s x_{sv} + \sum_{p \in P} \sigma_p x_{pv}. \quad (1)$$

При решении задачи должны быть соблюдены следующие ограничения:

– емкость памяти модуля L_v должна быть достаточной для выполнения приложений и хранения данных, размещенных в нем:

$$\sum_{s \in S} \sigma_s x_{sv} + \sum_{p \in P} \sigma_p x_{pv} \leq L_v, \forall v \in V, \quad (2)$$

– быстродействие процессора R_v модуля должно быть достаточным для выполнения с соответствующей частотой всех приложений, размещенных в модуле:

$$\beta_v \sum_{p \in P} v_p \rho_p x_{pv} \leq R_v, \forall v \in V \quad (3)$$

где β_v – коэффициент запаса на основе экспертных оценок.

Кроме того, каждое приложение должно выполняться на одном ВМ [5]:

$$\sum_{p \in P} x_{pv} = 1, \forall p \in P \quad (4)$$

Каждый массив должен храниться в памяти хотя бы одного модуля:

$$\sum_{v \in V} x_{sv} \geq 1, \forall s \in S \quad (5)$$

Выражения (1) – (5) определяют условия решения задачи размещения без учета архитектуры модулей КБО.

Тогда функциональную структуру комплекса удобно представить в виде неориентированного графа $G_F = (Y, Z)$, в котором каждому приложению или массиву ставится в соответствие вершина $y \in Y$, а информационной связи между ними – дуга $z \in Z$. Тогда аппаратная архитектура комплекса представляется неориентированным графом $G_S = (V, W)$, где вершины $v \in V$ соответствуют модулям, а ребра $w \in W$ – каналам связи между ними.

Для постановки задачи аппаратной реконфигурации требуется внести в модель комплекса дополнительные параметры. Пусть $k \in K$ – параметр критичности функции $y \in Y$, которая входит в состав неориентированного графа функциональной структуры $G_F = (Y, Z)$, где $z \in Z$ – дуга, соответствующая информационной связи между двумя функциями из множества Y . В соответствии с [5], множество K имеет пять членов k (в соответствии с количеством уровней гарантии разработки аппаратуры от А до Е). Положим, что $k_1 > k_2 > k_3 > k_4 > k_5$, т.е. k_1 соответствует уровню гарантии разработки аппаратуры А, k_2 – уровню В и т.д.

Таким образом, каждой вершине $y \in Y$ графа $G_F = (Y, Z)$ соответствует свой параметр критичности $k \in K$. Процесс назначения уровня критичности к какому-либо элементу комплекса ИМА является трудоемким и итеративным процессом с доказа-

тельством правильности присвоенного уровня и осуществляется, как правило, разработчиком. В данном примере для облегчения понимания задачи реконфигурации функции $y \in Y$ имеют уровни критичности, изображенные на рисунке 1.

Пусть $m \in M$ – количество отказавших модулей крейта ИМА. При этом аппаратная платформа КБО представляется неориентированным графом $G_S = (V, W)$, где вершины $v \in V$ соответствуют модулям, а ребра $w \in W$ – каналам связи между ними.

Таким образом, задача аппаратной реконфигурации состоит в перераспределении функций $y \in Y$ по исправным модулям $v \in V$ с учетом выбранного критерия оптимальности. При этом данное перераспределение должно учитывать критичность функций $k \in K$ и количество отказавших модулей $m \in M$.

Программная реконфигурация, так же, как и аппаратная, подразумевает прямую необходимость перераспределения функциональных приложений по аппаратной платформе. Так, например, если задачи были распределены по критерию минимума загрузки сети, то отказ какого-либо приложения приведет к уменьшению суммарного потока данных, либо оставит поток данных неизменным (если отказавшее приложение и приложение, соединенное с ним инцидентной дугой, находятся на одном модуле), что не требует проведения реконфигурации по заданному критерию оптимальности:

$$\min \left\{ \sum_{m \in Y} \sum_{n \in Y} v_{n,m} \sigma_{n,m} \right\} \leq \min \left\{ \sum_{m \in Y} \sum_{n \in Y} v_{n,m} \sigma_{n,m} \right\},$$

где m и n – натуральные числа, Y – множество всех функций до реконфигурации,

Y_r – множество всех функций после реконфигурации, $\sigma_{n,m}$ – размер массива данных, идущих по каналу связи от функции $n \in Y$ в функцию $m \in Y$, $v_{n,m}$ – частота обращения к данному массиву.

Но на практике выявлено, что повторное применение алгоритма распределения по алгоритму, описанному в [2] приводит к еще большему умень-

шению нагрузки на сеть. Это достигается за счет разрыва некоторых единичных связей между приложениями (отказавшими и рабочими), которые имели

большее значение суммарного потока данных, что позволило определить их в модули с другими рабочими приложениями, связь с которыми не нарушается.

Так же дело обстоит, если используется другой критерий, например, критерий минимума загрузки процессора каждого вычислительного модуля (ВМ). В случае отказа какого-либо приложения в системе освобождаются дополнительные процессорные ресурсы, что обязательно вызывает необходимость перераспределения.

Однако иногда выигрыш от проведения программной реконфигурации оказывается незначительным. Поэтому разработчик системы в системных алгоритмах и требованиях к ПО должен заранее определить набор функций, отказ которых должен приводить к перераспределению задач. Это очень трудоемкий процесс, поскольку количество различных вариаций отказов растет в геометрической прогрессии с ростом количества приложений. Поэтому необходимо либо разрабатывать специализированный САПР, либо проводить реконфигурацию при любом отказе, что может потребовать большого количества дополнительного времени для реконфигурации и сказаться на эффективности выполнения поставленной задачи.

Функциональная реконфигурация в общем случае проводится при отсутствии отказов. Ее смысл состоит в том, чтобы задействовать доступные вычислительные ресурсы аппаратной части комплекса и временные ресурсы ОСРВ для выполнения определенных задач с большей скоростью или точностью. Дополнительные ресурсы освобождаются в результате исключения тех или иных приложений из состава КБО при решении поставленной задачи. При этом поток данных в сети уменьшается, а время, доступное для выполнения приложений, увеличивается.

Основным отличием функциональной реконфигурации от реконфигураций других типов явля-

ется определенность конечных состояний КБО. Разработчик при проектировании комплекса должен составить перечень выполняемых приложений, соответствующих заданной полетной задаче, и осуществить их распределение по аппаратной платформе в соответствии с каким-либо алгоритмом. Сама же функциональная реконфигурация будет заключаться лишь в переходах между режимами, а закрепление приложений за модулями не потребует реализации какого-либо алгоритма непосредственно в полете. Исключение могут составить алгоритмы физического переключения аппаратной платформы в процессе реконфигурации при переходах между режимами, но в данной статье такие алгоритмы не рассматриваются.

Однако на практике может возникнуть такая ситуация, когда потребуется одновременно провести несколько типов реконфигурации. В этом случае распределение приложений только лишь в соответствии с режимами полета теряет смысл, поскольку оно не учитывает возникшие отказы. Поэтому алгоритмы функциональной реконфигурации должны учитывать не только структуру комплекса и набор приложений, но и результаты программной и аппаратной реконфигурации.

Чтобы определить задачу смешанной реконфигурации в дополнение к описанным выше параметрам, относящимся к аппаратной реконфигурации, добавляются новые. Для выполнения различных задач в полете определим множество режимов Q . Количество режимов Q определяется разработчиком, задается в техническом задании и заранее известно. Перечень функций $y_q \in Y_q$ выполняющихся в режиме $q \in Q$ так же определен, и представляет собой неориентированный граф функциональной структуры $G_{Fq} = (Y_q, Z_q)$, где $z \in Z_q$ – дуга, соответствующая информационной связи между двумя функциями из множества Y_q .

При этом объединение всех множеств Y_q дает нам исходное множество функций Y :

$$\bigcup_q Y_q = Y$$

Соответственно объединение графов, соответствующих режимам полета, даст нам исходный граф

функциональной структуры G_F :

$$\bigcup_q G_{Fq} = G_F.$$

Таким образом, задача смешанной реконфигурации состоит в перераспределении функций $y \in Y$ по исправным модулям $v \in V$ с учетом выбранного критерия оптимальности. При этом данное перераспреде-

ление должно учитывать критичность функций $k \in K$, количество отказавших модулей $m \in M$ и приложений $n \in N$, а также установленный режим полета $q \in Q$.

2. Алгоритмы реконфигурации КБО ИМА

- В качестве примера рассмотрен КБО вертолета. В общем случае он должен обеспечивать выполнение следующих функций:
1. Решение задач навигационного обеспечения;
 2. Вертолетовождение по запрограммированному маршруту;
 3. Формирование и отображение полетно-навигационной информации (ПНИ);
 4. Пилотирование и управление полетом;
 5. Круглосуточная обзорная система;
 6. Оперативный ручной ввод заданных значений параметров полета;
 7. Отсчет и отображение текущего и полетного времени;
 8. Ручная и автоматическая настройка радиотехнических систем навигации и посадки;
 9. Обеспечение взаимодействия с радиолокаторами управления воздушным движением;
 10. Передача информации в обеспечение автоматического зависимого наблюдения;
 11. Контроль состояния силовой установки и общевертолетного оборудования;
 12. Управление общевертолетным оборудованием;
 13. Управление системой электропитания бортового оборудования;
 14. Формирование и регистрация массива полетной информации в бортовом регистраторе;
 15. Функции комплекса средств связи;
 16. Передача радиосигналов бедствия;
 17. Полетный автоматический контроль исправности комплекса с отображением результатов контроля.
 18. Управление вооружением;
 19. Управление системами постановки помех;
 20. Многоканальная система наведения.

Предполагается, что КБО функционирует в двух режимах – режим крейсерского полета и режим боя. Функциональные графы G_{F1} и G_{F2} данного КБО для крейсерского полета и боя соответственно представлены на рисунке 1.

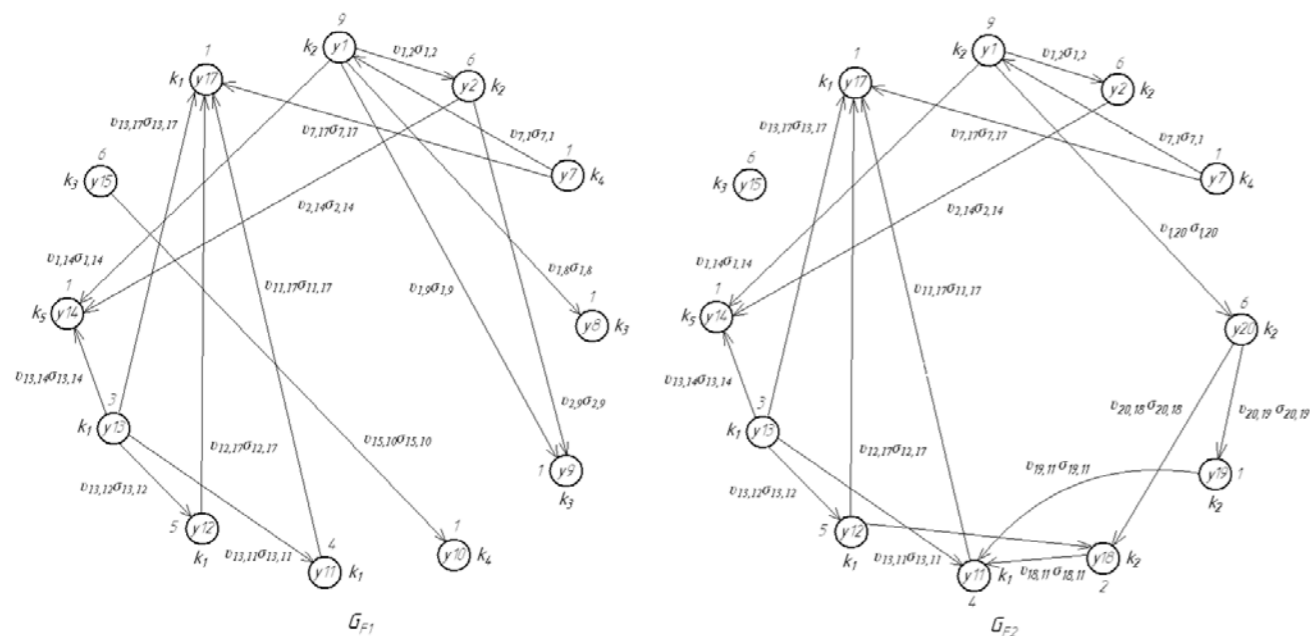


Рис. 1. Функциональные графы вычислительной системы КБО

Порядковые номера вершин графа соответствуют порядковым номерам перечисленных выше выполняемых функций, цифровые значения – требуемым ресурсам процессора, параметры k – уровням критичности. При этом задачи под номерами 18–20 характерны только для режима боя, а задачи 8–10 – только для крейсерского полета. При этом учитыва-

ваются функции, которые требуют реализации на специальном вычислителе или вне крейта ИМА:

6. Оперативный ручной ввод заданных значений параметров полета;
16. Передача сигналов бедствия.

Результаты работы разработанных алгоритмов распределения ресурсов и динамической реконфигурации, подробно описанных в [2] и других работах автора, представлены на рисунках 2–4.

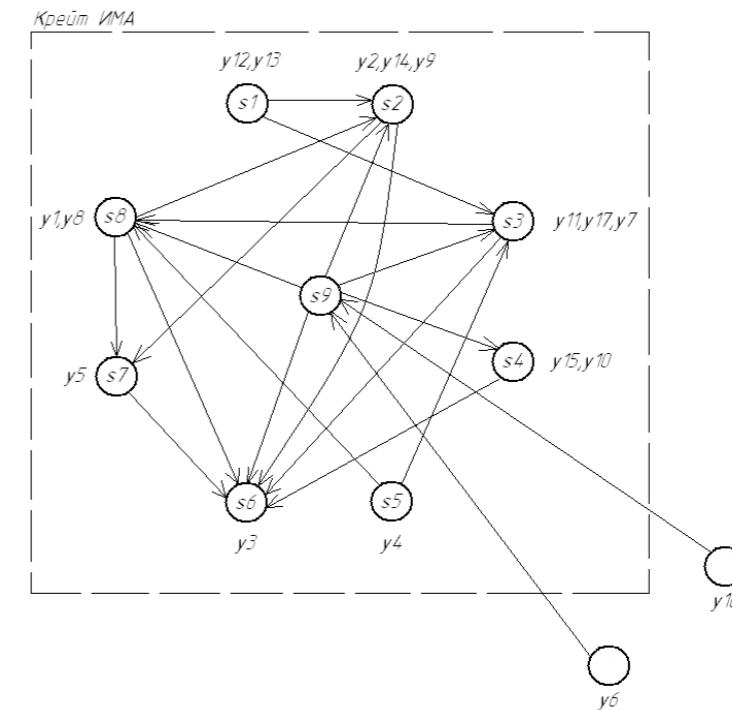
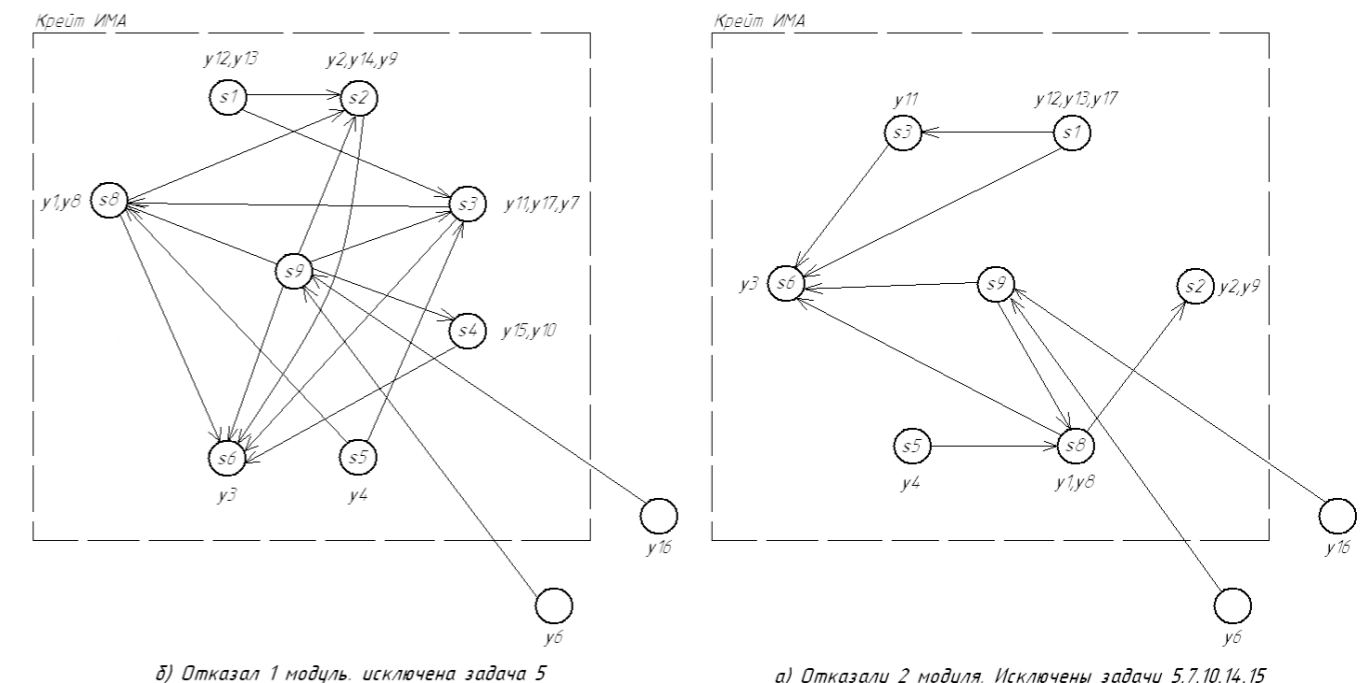
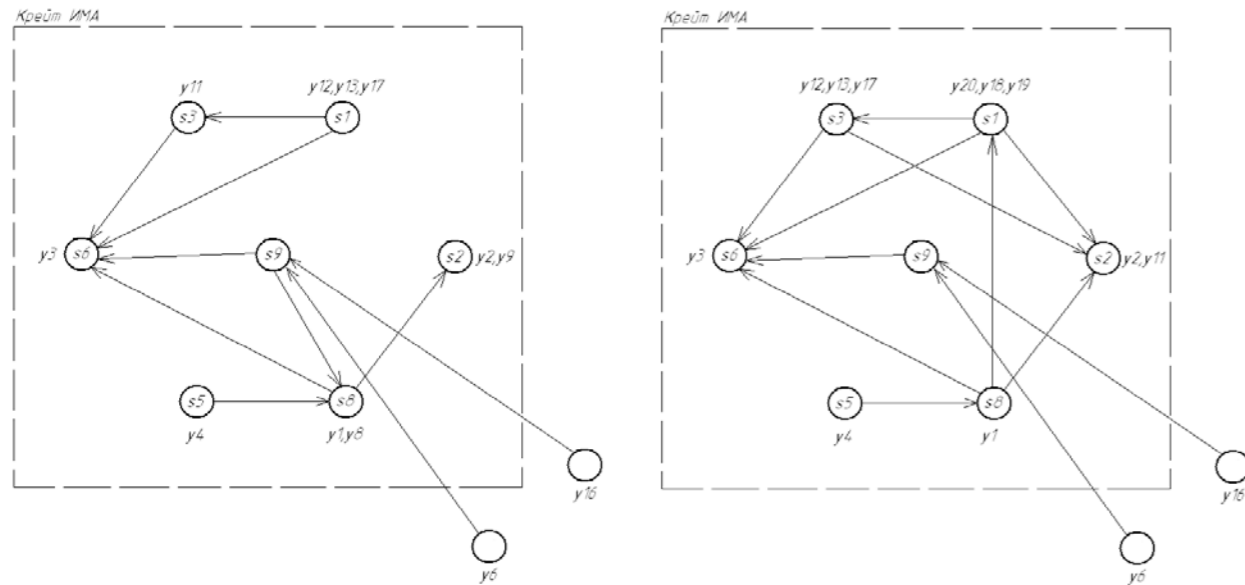


Рис. 2. Граф аппаратной структуры системы ИМА с распределенными функциональными задачами по критерию минимума загрузки сети



а) Отказали 2 модуля. Исключены задачи 5, 7, 10, 14, 15
б) Отказал 1 модуль, исключена задача 5

Рис. 3. Работа алгоритма аппаратной реконфигурации при отказах одного и двух модулей крейта



а) Крейсерский полет. Отказали 2 модуля. Исключены задачи 5,7,10,14,15

б) Режим боя. Отказали 2 модуля. Исключены задачи 5,7,14,15

Рис. 4. Работа алгоритма смешанной реконфигурации в различных режимах полета

Вершины s_1-s_5, s_7, s_8 представляют собой базовые вычислительные модули (ВМ), вершина s_6 – модуль графической обработки (МГО), s_9 – модуль-концентратор сигналов (МКС).

3. Оценка эффективности разработанных алгоритмов реконфигурации

Оценка эффективности разработанных алгоритмов проводилась путем анализа деревьев неисправности КБО в соответствии с Р4761 [6]. Анализ дерева неисправности – дедуктивный анализ отказов, который сосредотачивается на одном конкретном нежелательном событии и который позволяет определить причины этого события. Анализ дерева неисправности – “нисходящая” процедура оценки системы, в которой формируется и оценивается качественная, а затем и количественная модель для отказного события.

Каждое конечное событие дерева неисправности непосредственно является полным отказом одного функционального узла, либо определенным видом отказа функционального узла. Вероятность отказа функционального узла определяется в соответствующем расчете надежности.

Для КБО, состоящего из двух крейтов ИМА, структура которых изображена на рисунке 2, конечным (базовым) событием дерева будет отказ какого-либо ВМ, МГО или МКС.

Поскольку программное обеспечение (ПО)

таких комплексов разрабатывается по уровню А в соответствии с КТ-178В [7], вероятность необнаруженной ошибки в коде, приводящей к соответствующему событию в дереве неисправности считается событием практически невероятным ($1 \times E-09$). Поэтому данные виды отказов при анализе не включались в состав дерева.

Чтобы оценить эффективность алгоритмов реконфигурации, были построены и проанализированы деревья неисправности для классической федеративной архитектуры с двойным резервированием и архитектуры ИМА, состоящей из одного крейта с резервированным коммутатором.

Дерево неисправности для федеративной архитектуры (для одного блока) изображено на рисунке 5.

Поскольку все базовые ВМ идентичны друг другу, а отличия их от МКС и МГО в части интенсивности отказов минимальны (одного порядка), то интенсивность отказов каждого модуля λ_s принимается равной $1 \times E-006$. Поскольку коммутатор является самым уязвимым местом во всей системе, то он, как правило, резервируется дополнительно или проектируется из элементов повышенной надежности. Поэтому при расчетах его интенсивность отказов была принята $5 \times E-007$. Полный отказ федеративной системы происходит при отказе хотя бы одного модуля.

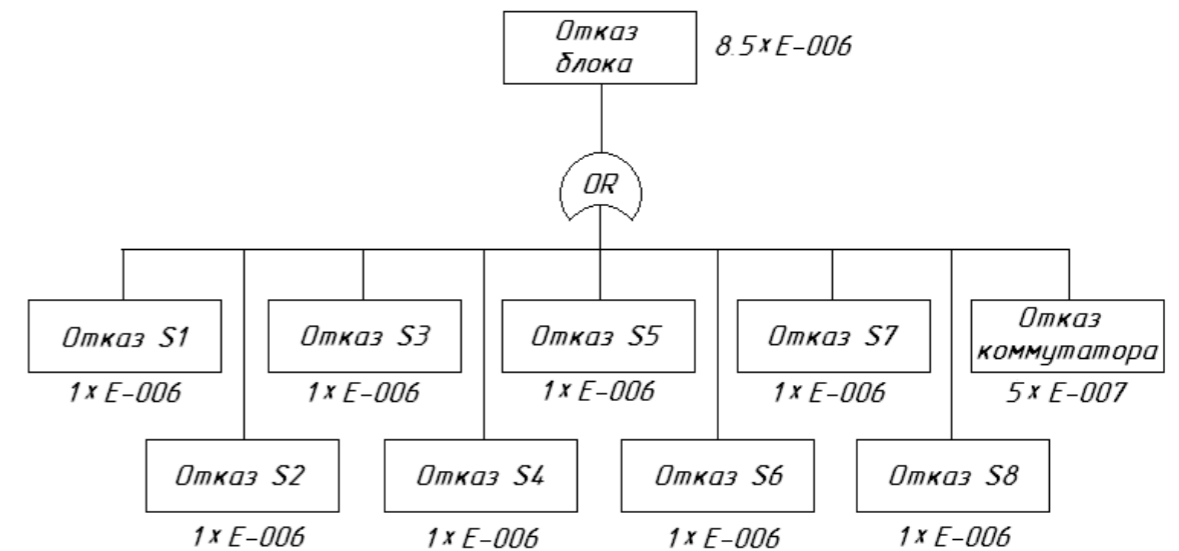


Рис. 5 Дерево неисправностей блока федеративной архитектуры

Дерево неисправности для архитектуры ИМА (для одного крейта) изображено на рисунке 6.

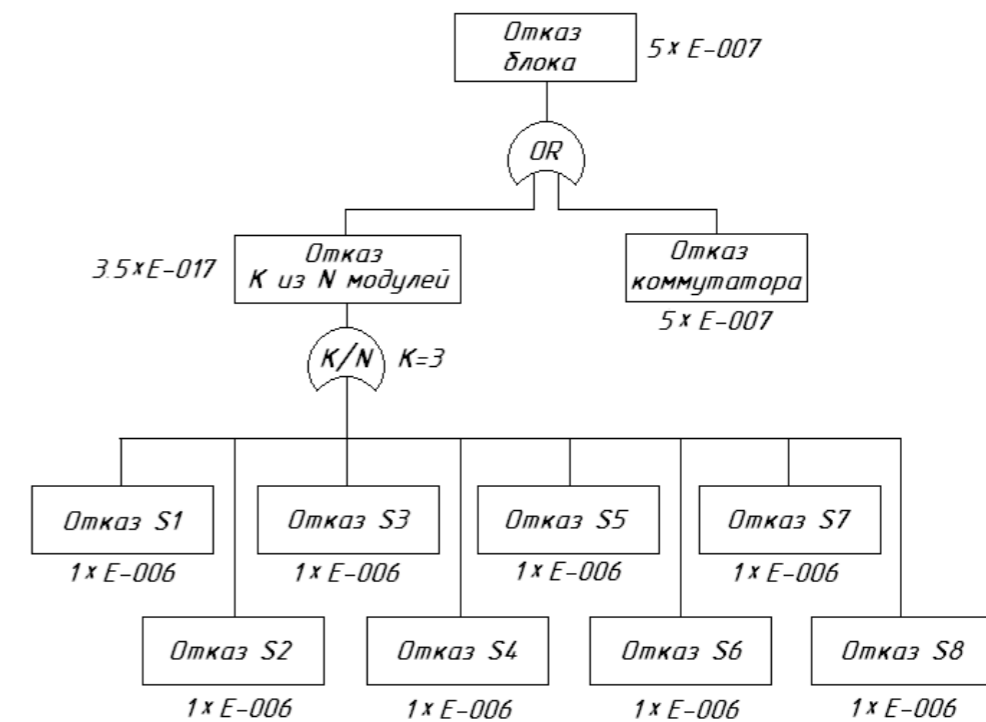


Рис. 6 Дерево неисправностей блока архитектуры ИМА

Отказ блока с архитектурой ИМА, аппаратная структура которого представлена на рисунке 2 и который поддерживает возможность реконфигурации по разработанным алгоритмам, произойдет, если откажут хотя бы три ВМ, поскольку для выполнения приложений $y \in Y$, которые относятся к уровням критичности k_1 и k_2 , необходимо четыре ВМ, в то время как остальные задачи исключаются из состава КБО при возникновении отказов (см. таблицу 1).

Таблица 1

Уровни критичности функций по КТ-254 для режима крейсерского полета вертолета

$k \in K$	Функции $y \in Y$					
k_1	13	11	12	17	4	3
k_2	2	1				
k_3	8	9	15			
k_4	10	5	7			
k_5	14					

Построения деревьев неисправностей и их расчеты проводились при помощи программы RAM Commander с параметрами относительной величины отсечения $1 \times E-006$, точности оптимизации 0,001 и продолжительности полета 1 ч. При этом для модулей задавались значения интенсивностей отказов λS при экспоненциальном законе распределения отказов.

Как видно из рисунков 5 и 6, вероятность отказа реконфигурируемого крейта ИМА более чем в 15 раз превосходит аналогичный показатель для блока федеративной архитектуры ($5 \times E-007$ у ИМА против $8,5 \times E-006$ у федеративной архитектуры). Более того расчеты показали, что даже федеративная система с двойным резервированием блоков не обладает лучшими показателями надежности, чем один реконфигурируемый крейт с резервированным коммутатором ($2,5 \times E-013$ у ИМА против $7,2 \times E-011$ у федеративной архитектуры).

Таким образом, реконфигурируемый крейт ИМА не только гораздо более надежен, но и более экономичен с точки зрения ресурсов аппаратуры (количество элементов, вес) и ремонтпригодности, так как его основу составляют идентичные ВМ.

Заключение

Для усовершенствования разработанных алгоритмы реконфигурации КБО ИМА необходимо решить задачу оптимального распределения приложений одновременно по множеству критериев (минимальная загрузка сети передачи данных, минимальная нагрузка процессоров, минимальная нагрузка памяти и т. д.). При этом возможно разде-

ление многомерной функциональной задачи на отдельные подзадачи, которые требуют обеспечения каких-то частных функционалов оптимальности. То есть какая-то часть системы может быть критична к задержкам передачи данных, а какая-то – к нагрузке процессора.

Повышение эффективности разработанных алгоритмов так же может быть достигнуто путем учета специфики контроля приложений операционной системой реального времени (ОСРВ) и типа используемой мультипроцессорной системы [8]. Наложив дополнительные ограничения можно осуществить частичную динамическую реконфигурацию, при которой перераспределяются не все функциональные приложения, а только те, которые были запущены на вышедших из строя модулях. Но реализация такого функционала сильно зависит от архитектуры применяемой ОСРВ.

Предлагаемые на сегодняшний день многопроцессорные вычислительные системы с динамически перестраиваемой архитектурой, подробно описанные в работах [9–11], не могут быть применены к рассматриваемым системам ИМА, которые требуют именно динамической реконфигурации во время полета. Создание частных алгоритмов реконфигурации может служить основой для разработки эффективных реконфигурируемых систем, сохраняющих свою работоспособность при множественных аппаратных отказах. Однако данные алгоритмы требуют наличия дополнительных специальных средств (как программных так и аппаратных) для их реализации.

Список литературы

1. Чуянов Г.А. Перспективы развития комплексов бортового оборудования на базе интегрированной модульной авионики [Текст] / Г.А. Чуянов, В.В.Косьянчук, Н.И. Сельвесюк // Известия ЮФУ. Технические науки. – 2013. – № 3. – С. 55–62.
2. Conmy Philippa. High level failure analysis for Integrated Modular Avionics [Text]/ Philippa Conmy, John McDermid // 6th Australian Workshop on Safety Critical Systems and Software, 2001.

3. Дегтярев А.Р. Разработка перспективных динамически реконфигурируемых систем интегрированных комплексов бортового оборудования [Текст] / А.Р. Дегтярев // Сборник тезисов докладов конгресса молодых ученых. – Выпуск 1. – СПб.: Университет ИТМО, 2014. – 232 с.

4. Дегтярев А.Р. Алгоритм распределения задач в многопроцессорных омплексах интегрированной модульной авионики [Текст] / А.Р. Дегтярев, Г.В. Медведев //Автоматизация процессов управ-

ления – 2014. – № 1. – С. 79–84.

5. Руководство по гарантии конструирования бортовой электронной аппаратуры КТ-254 [Текст]. – М.: Авиаиздат, 2011. – 86 с.

6. Руководство 4761 по методам оценки безопасности систем и бортового оборудования воздушных судов гражданской авиации [Текст]. – М.: ОАО Авиаиздат, 2010. – 264 с.

7. Квалификационные требования КТ-178В “Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники” [Текст]. – М.: НИИАО, 2002. – 105 с.

8. Таненбаум Э. Современные операционные системы [Текст] / Э. Таненбаум. – 3-е изд. – СПб.: Питер, 2010. – 1120 с.: ил.

9. Реконфигурируемые мультимикровейерные вычислительные структуры [Текст] / И.А. Каляев [и др.] – Ростов н/Д.: Издательство ЮНЦ РАН, 2008. – 393 с.

10. Каляев А.В. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений [Текст] / И.А. Каляев, И.И. Левин. – М.: Янус-К, 2003. – 380 с.

11. Беседин И.В. Семейство базовых модулей для построения реконфигурируемых многопроцессорных вычислительных систем со структурнопроцедурной организацией вычислений [Текст] / И.В. Беседин [и др.] // Научный сервис в сети Интернет: технологии распределенных вычислений: материалы всероссийской научной конференции. – М.: Издательство Московского университета, 2006. – С. 47–49.

References

1. Chujanov G.A. Perspektivy razvitiya kompleksov bortovogo oborudovaniya na baze integrirovannoj modul'noj avioniki [Tekst] / G.A. Chujanov, V.V.Kos'janchuk, N.I. Sel'vesjuk // Izvestija JuFU. Tehnicheskie nauki. – 2013. – № 3. – P. 55–62.

2. Conmy Philippa. High level failure analysis for Integrated Modular Avionics [Text]/ Philippa Conmy,

John McDermid // 6th Australian Workshop on Safety Critical Systems and Software, 2001.

3. Degtjarev A.R. Razrabotka perspektivnyh dinamicheski rekonfiguriruemyh sistem integrirovannyh kompleksov bortovogo oborudovaniya [Tekst] / A.R. Degtjarev // Sbornik tezisov dokladov kongressa molodyh uchenyh. – Vypusk 1. – SPb.: Universitet ITMO, 2014. – 232 p.

4. Degtjarev A.R. Algoritm raspredelenija zadach v mnogoprocessornyh ompleksah integrirovannoj modul'noj avioniki [Tekst] / A.R. Degtjarev, G.V. Medvedev //Avtomatizacija processov upravlenija – 2014. – № 1. – P. 79–84.

5. Rukovodstvo po garantii konstruirovaniya bortovoj jelektronnoj apparatury KT-254 [Tekst]. – М.: Aviaizdat, 2011. – 86 p.

6. Rukovodstvo 4761 po metodam ocenki bezopasnosti sistem i bortovogo oborudovaniya vozdushnyh sudov grazhdanskoj aviicii [Tekst]. – М.: ОАО Aviaizdat, 2010. – 264 p.

7. Kvalifikacionnye trebovanija KT-178B “Trebovanija k programnomu obespecheniju bortovoj apparatury i sistem pri sertifikacii aviacionnoj tehniki” [Tekst]. – М.: NPIAO, 2002. – 105 p.

8. Tanenbaum Je. Sovremennye operacionnye sistemy [Tekst] / Je. Tanenbaum. – 3-e izd. – SPb.: Piter, 2010. – 1120 s.: il.

9. Rekonfiguriruemye mul'tikonvejernye vychislitel'nye struktury [Tekst] / I.A. Kaljaev [i dr.] – Rostov n/D.: Izdatel'stvo JuNC RAN, 2008.– 393 p.

10. Kaljaev A.V. Modul'no-narashhivaemye mnogoprocessornye sistemy so strukturno-procedurnoj organizaciej vychislenij [Tekst] / I.A. Kaljaev, I.I. Levin. – М.: Janus-K, 2003. – 380 p.

11. Besedin I.V. Semejstvo bazovyh modulej dlja postroeniya rekonfiguriruemyh mnogoprocessornyh vychislitel'nyh sistem so strukturnoprocedurnoj organizaciej vychislenij [Tekst] / I.V. Besedin [i dr.] // Nauchnyj servis v seti Internet: tehnologii raspredelennyh vychislenij: materialy vsrossijskoj nauchnoj konferencii. – М.: Izdatel'stvo Moskovskogo universiteta, 2006. – P. 47–49.