

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи

Дегтярев Алексей Робертович

**МОДЕЛИ И АЛГОРИТМЫ УПРАВЛЕНИЯ КОНФИГУРАЦИЕЙ СИСТЕМ
ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИАНИКИ**

Специальность 05.13.05 – Элементы и устройства вычислительной
техники и систем управления

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель:

д. т. н., доцент Киселев Сергей Константинович

Ульяновск – 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 АНАЛИЗ КОМПЛЕКСОВ ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИОНИКИ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ	12
1.1 Аналитический обзор и анализ архитектуры существующих КБО ИМА	12
1.2 Современные стандарты проектирования КБО ИМА	37
1.3 Современные методы проектирования КБО ИМА	40
1.4 Выводы	49
2 РАЗРАБОТКА И ИССЛЕДОВАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ КОМПЛЕКСА БОРТОВОГО ОБОРУДОВАНИЯ ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИОНИКИ	51
2.1 Математическое описание состава КБО ИМА	53
2.2 Математическое описание структуры КБО ИМА	56
2.3 Математическая модель КБО ИМА. Алгоритм распределения функциональных задач	62
2.4 Выводы	80
3 АЛГОРИТМЫ ДИНАМИЧЕСКОЙ РЕКОНФИГУРАЦИИ КБО ИМА	82
3.1 Механизмы динамической реконфигурации	83
3.2 Алгоритм аппаратной динамической реконфигурации	92
3.3 Программная и смешанная реконфигурации. Алгоритм смешанной динамической реконфигурации	99
3.4 Выводы	111
4 ИССЛЕДОВАНИЕ РЕКОНФИГУРИРУЮЩЕГОСЯ КБО ИМА	113
4.1 Оценка результатов работы алгоритмов оптимального распределения и реконфигурации	113
4.2 Оценка безопасности и надежности реконфигурирующегося КБО ИМА	118
4.3 Методика предварительного определения конфигурации КБО ИМА	134
4.4 Выводы	138
ЗАКЛЮЧЕНИЕ	141
СПИСОК ЛИТЕРАТУРЫ	147
СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА	158
Приложение А Дерево неисправностей распределенной системы ИМА	161
Приложение Б Дерево неисправностей реконфигурирующейся системы ИМА	162
Приложение В Акт внедрения результатов диссертационной работы	163

ВВЕДЕНИЕ

Комплексы бортового оборудования (КБО) на основе концепции интегрированной модульной авионики (ИМА), пришедшие на смену традиционной федеративной архитектуре, представляют собой следующий шаг в развитии авиационного приборостроения и неразрывно связаны с созданием новых типов летательных аппаратов (ЛА), которые обладают улучшенными летными характеристиками, и к которым предъявляются повышенные требования по надежности и унификации структурных элементов.

Необходимость наращивания функционала КБО с одновременным снижением их стоимости и эксплуатационных расходов, а также стремление к глубокой интеграции аппаратных, программных и алгоритмических ресурсов определяют следующие направления развития ИМА:

- создание унифицированных открытых отказоустойчивых архитектур КБО с применением перспективных высокоскоростных сетевых интерфейсов;
- дальнейшая унификация аппаратуры с целью снижения номенклатуры разрабатываемых изделий и сокращения сроков выпуска конструкторской документации (КД), улучшение массогабаритных характеристик, увеличения общей производительности системы, ее надежности и отказоустойчивости;
- расширение функционала КБО для обеспечения безопасности и повышения эффективности выполнения поставленных задач;
- создание эффективных методов и средств встроенного контроля с целью повышения отказоустойчивости и обеспечения простоты технического обслуживания, в том числе и в полевых условиях;
- разработка и внедрение динамического управления конфигурацией для обеспечения живучести КБО и выполнения поставленных задач при возникновении одиночных или множественных отказов.

Одним из ключевых вопросов является разработка алгоритмов функционирования и средств реализации отказоустойчивого КБО, одним из основных свойств которого является использование активной системы безопасности, позволяющей в

случае возникновения в комплексе частичных отказов изменять его структуру для частичного или полного сохранения функций. Построение такого комплекса возможно путем динамической реконфигурации его ресурсов, что позволит минимизировать резервирование и обеспечит максимально возможный уровень безопасности и эффективности завершения полета или его дальнейшего выполнения в случае возникновения отказов.

Для комплексного решения данного вопроса необходимо разработать информационные, логические и математические модели системы ИМА, алгоритмы встроенного контроля состояния системы, основанные на этих моделях, а также алгоритмы динамического управления конфигурацией системы в процессе функционирования комплекса.

Достижению поставленной задачи во многом способствует быстрое развитие элементной базы, в том числе процессоров и программируемых логических интегральных схем (ПЛИС), которые активно используются при проектировании аппаратуры КБО. Новые микросхемы отличаются не только расширенным функционалом, но и улучшенными временными, надежностьвыми и массогабаритными характеристиками, что позволяет осуществлять на их основе глубокую интеграцию аппаратной части проектируемых комплексов.

Исследования в области создания систем ИМА велись с начала 90-х годов прошлого века. Основные результаты работы по проектированию аппаратной части комплексов нашли отражение в работах Грабовски Г., Балсера Б., Форстера М., Алена Р., Оссенфорта Дж., Блэка Р. [1-3]. В них обсуждались проблемы создания надежных конструктивных решений для реализации модулей, в том числе и на основе коммерческих компонентов, проблемы организации высокоскоростной сети передачи данных (с использованием Avionics Full-Duplex Switched Ethernet - AFDX) [4], а также вопросы перехода от традиционной федеративной архитектуры к концепции ИМА.

Проблемами программного обеспечения и моделирования комплексов занимались Гаматье А., Брунетт К., Готье Т., Конми Ф., Дермид Дж., Паркинсон П., Гасперони Ф., Киннан Л., Миллер С., Коффер Д. и др. [5-14]. Для данных работ

характерна концентрация на проблемах проектирования ПО с высокими показателями безопасности (*safety critical*), планирования и распределения ресурсов комплекса, а также тестирования на основе моделей.

Вопросами операционных систем реального времени (ОСРВ) во встраиваемых системах занимались Бэкон. Дж. и Харрис Т. [15], Таненбаум Э.С. [16], Вудхалл А.С. [17].

Развитие методологии проектирование КБО ИМА в России началось только в начале 2000-х годов, когда первые зарубежные комплексы уже внедрялись в серийное производство. Наибольший вклад в исследование ИМА вложили Гатчин Ю.А., Видин В.Б., Жаринов И.О., Жаринов О.О., Парамонов П.П., Виноградов П.С. [18-23], которые предложили математические модели и методы автоматизированного проектирования аппаратных средств ИМА, Джанджгава Г.И., Герасимов Г.И. и Евгенов А.В., Турчак А.А. и др. [24-29], которые предложили концепции создания КБО ИМА нового поколения. Проблемами системного обмена и распределения ресурсов в комплексах занимались Богатырев В.А., Голубев И.Ю., Беззубов В.Ф. [30, 31], Афраймович Л.Г., Прилуцких М.Х. [32]. Работы Буздalова Д.В. Зеленова С.В., Корныхина Е.В., Петренко А.К. и других авторов посвящены инструментальным средствам проектирования систем ИМА [33, 34].

Отдельно стоит отметить авторов, которые занимались вопросами создания реконфигурируемых многопроцессорных систем. Среди них Каляев И.А., Каляев А. В., Левин И.И., Беседин И.В., Дмитриенко Н.Н., Семерников Е.А., Чуянов Г.А., Косьянчук В.В., Севельсюк Н.И. [35-39].

В настоящее время работы по проектированию архитектуры КБО для перспективных ЛА осуществляется рядом предприятий Российской Федерации: АО «Ульяновское конструкторское бюро приборостроения», ФГУП «СПб ОКБ «Электроавтоматика» им. П.А. Ефимова», ВНИИРА (Санкт-Петербург), МИЭА (г. Москва), НПО «Полет» (Нижний Новгород), ФГУП «НИИ Авиационных систем» (г. Москва), ОАО «РПКБ «Раменское» (г. Москва) и др. [40]. Достигнутые в настоящее время результаты разработки защищены патентами [41-45].

К числу государственных программ, направленных на модернизацию существующих и разработку новейших КБО с использованием перспективных технологий относятся:

- государственная программа Российской Федерации «Развитие авиационной промышленности на 2013-2025 годы», утвержденная постановлением Правительства Российской Федерации от 15 апреля 2014 г. № 303 «Об утверждении государственной программы Российской Федерации «Развитие авиационной промышленности на 2013-2025 годы»;

- государственная программа Российской Федерации «Развитие электронной и радиоэлектронной промышленности на 2013-2025 годы», утвержденная распоряжением Правительства Российской Федерации от 15 декабря 2012 г. № 2396-р;

- федеральная целевая программа «Развитие оборонно-промышленного комплекса Российской Федерации на 2011-2020 годы», утвержденная Распоряжением Правительства Российской Федерации от 26.05.2011 № 912-рс.

Новейшими стандартами проектирования систем ИМА являются:

- руководство Р-297 по вопросам разработки и квалификации интегрированной модульной авионики, введенное в действие Авиационным Регистром Межгосударственного Авиационного Комитета (АР МАК) 20 марта 2015 года директивным письмом № 05-2015;

- руководство 4761 по методам оценки безопасности систем и бортового оборудования воздушных судов гражданской авиации;

- руководство 4754 по процессам сертификации высокоинтегрированных сложных бортовых систем воздушных судов гражданской авиации.

Таким образом, исследования принципов построения и методологии проектирования систем на базе ИМА на сегодняшний день являются **актуальными**, способствуют повышению конкурентоспособности отечественных производителей и ускорению научно-технического прогресса, что позволит вывести проектирование авионики на качественно новый уровень.

Цель диссертационной работы состоит в развитии методов проектирования систем ИМА в соответствии с основными положениями концепции ИМА, в разработке моделей и алгоритмов оптимального распределения ресурсов КБО и алгоритмов динамического управления структурой системы при возникновении отказов, которые позволят обеспечить требуемые показатели надежности проектируемых комплексов.

В соответствии с обозначенной целью в работе поставлены и решены следующие задачи:

1. Определены структурные особенности комплексов ИМА, обеспечивающие возможность управления их конфигурацией.

2. Проанализирована возможность непрерывного контроля состояния комплекса и изменения его структуры в процессе выполнения задач без необходимости проведения дополнительных сервисных операций.

3. Проанализированы основные виды изменения конфигурации систем ИМА: аппаратная динамическая реконфигурация, программная динамическая реконфигурация, функциональная динамическая реконфигурация, смешанная динамическая реконфигурация.

4. Разработана единая математическая модель аппаратной и программной структур системы ИМА.

5. Разработан алгоритм определения базовой конфигурации системы ИМА (распределения программных ресурсов по аппаратной платформе системы с учетом выбранного критерия оптимальности – минимума загрузки сети передачи данных).

6. Разработаны алгоритмы динамического изменения конфигурации системы ИМА для различных условий ее функционирования.

7. Проведены вычислительные эксперименты и проверка разработанных алгоритмов на макете многопроцессорной вычислительной системы.

8. Проведена оценка эффективности разработанных моделей и алгоритмов для достижения требуемых показателей надежности при проектировании образца системы ИМА, проходящей этап межведомственных испытаний.

Положения, выносимые на защиту, обладающие научной новизной:

1. Единая математическая модель, объединяющая аппаратную и программную составляющие системы ИМА, отличающаяся тем, что кроме связей между аппаратной и программной частями системы в ней учитываются особенности архитектуры аппаратной и организации программной частей, что позволяет использовать ее для разработки алгоритмов управления конфигурацией систем ИМА с обеспечением требуемого уровня надежности системы в целом.

2. Алгоритм определения базовой конфигурации системы ИМА на основе заданного критерия оптимальности с учетом критичности функциональных приложений и требуемого уровня гарантии конструирования аппаратуры.

3. Алгоритмы динамического изменения конфигурации системы ИМА, позволяющие обеспечивать требуемый уровень надежности при различных условиях ее функционирования.

4. Методика определения конфигурации системы ИМА, основанная на разработанных моделях и алгоритмах управления конфигурацией и обеспечивающая требуемые показатели надежности на различных этапах ее проектирования в соответствии с современными стандартами.

Объектом исследования в диссертационной работе является проектирование систем ИМА, **предметом исследования** являются разработанная математическая модель реконфигурирующейся системы ИМА и алгоритмы распределения функциональных задач и динамической реконфигурации.

Методы исследования базируются на теории алгоритмов, теории графов, теории надежности.

Новизна научных положений состоит в следующем:

1. В единой математической модели системы ИМА описаны состав и структура как аппаратной, так и программной составляющих системы, а также, кроме связей между аппаратной и программной частями системы, в ней учтены особенности архитектуры аппаратной и организации программной частей.

2. Алгоритм определения базовой конфигурации системы ИМА на основе заданного критерия оптимальности (представляющий собой модифицированный

вариант алгоритма решения задачи об упаковке) позволяет учитывать критичность функциональных приложений и требуемый уровень гарантии конструирования аппаратуры.

3. Выявлены и показаны идентичность и отличия решений задач аппаратной, программной и смешанной реконфигурации системы ИМА, с учетом чего предложены взаимосвязанные алгоритмы динамического изменения конфигурации системы, позволяющие обеспечивать требуемый уровень ее надежности при различных условиях функционирования.

Практическая значимость работы и реализация ее результатов:

1. Алгоритм распределения программных приложений по минимуму загрузки сети передачи данных в КБО ИМА и соответствующая ему параметрическая поверхность вариантов построения системы позволяют с достаточной точностью на начальном этапе проектирования оценить модульный состав разрабатываемого реконфигурирующегося крейта ИМА.

2. Разработанные модель и алгоритмы определения конфигурации КБО ИМА позволяют получать варианты системы, надежность которых в среднем на порядок выше, чем у систем федеративной архитектуры, что подтверждается анализом деревьев неисправностей, проведенным в соответствии с «Руководством R-4754 по сертификации сложных бортовых систем воздушных судов гражданской авиации» (SAE ARP-4754) и «Руководством R-4761 по методам оценки безопасности систем и бортового оборудования воздушных судов гражданской авиации» (SAE ARP-4761).

3. Предложенные алгоритмы изменения конфигурации системы ИМА (аппаратной, программной, смешанной) и механизм ее реализации через определение таблиц реконфигурации для основных возможных складывающихся режимов полета и технического состояния системы позволяют поддерживать работоспособность КБО на уровне, требуемом нормами летной годности, обеспечивать функционирование системы ИМА в аварийном режиме для совершения экстренной посадки.

4. Разработанные модель и алгоритмы определения конфигурации КБО ИМА, обобщенная методика их использования позволяют организовывать процесс проектирования системы ИМА в соответствии с требованиями «Руководства по гарантии конструирования бортовой электронной аппаратуры КТ-254» (RTCA DO-254/EUROCAE ED-80 «Design Assurance Guidance for Airborne Electronic Hardware») и квалификационными требованиями КТ-178В «Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники» (RTCA/DO-178В «Software considerations in airborne systems and equipment certification»).

Результаты работы получены автором при выполнении в интересах гражданской авиации РФ составных частей ОКР:

- «Создание образцов комплекса унифицированных базовых элементов бортового оборудования открытой архитектуры на основе интегральной модульной авионики» по теме «Разработка курсоверткали», исполнитель – АО «УКБП», 2012-2015 гг.;

- «Разработка модуля воздушных данных МВД», исполнитель – АО «УКБП», 2012-2015 гг.

- «Разработка платформы ИМА ИМ-21-2», исполнитель – АО «УКБП», 2013-2015 гг.

Достоверность результатов. Достоверность проведенных теоретических исследований обеспечивается математическим обоснованием предлагаемых моделей и алгоритмов, а также успешным их применением при решении практических задач по разработке КБО.

Апробация работы. Основные положения и результаты диссертационной работы докладывались и обсуждались на:

- III и IV Всероссийских конгрессах молодых ученых НИУ ИТМО (г. Санкт-Петербург, 2014 и 2015 гг.);

- всероссийской научно-технической конференции «XII Научные чтения по авиации, посвященные памяти Н.Е. Жуковского» (г. Москва, 2015 г.);

- II Всероссийской научно-практической конференции «Прикладные информационные системы-2015» (г. Ульяновск, 2015 г.);
- VI Всероссийской научно-технической конференции аспирантов, студентов и молодых ученых «Информатика и вычислительная техника» (г. Ульяновск, 2014 г.);
- всероссийской молодежной школе-семинаре «Актуальные проблемы информационных технологий, электроники и радиотехники - 2015» (ИТЭР – 2015, Южный федеральный университет, г. Ростов-на-Дону, г. Таганрог, 2015 г.);
- всероссийской научно-технической конференции «Авиационные приборы и измерительно-вычислительные комплексы ИВК-2013» (г. Ульяновск, 2013 г.);
- научно-технических конференциях профессорско-преподавательского состава Ульяновского государственного технического университета «Вузовская наука в современных условиях» (Ульяновск, 2012-2016 гг.).

1 АНАЛИЗ КОМПЛЕКСОВ ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИОНИКИ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

Системы на базе концепции ИМА на сегодняшний день являются предметом большого числа исследований. Востребованность данного вида комплексов обусловлена не только их гибкостью и возможностью установки на различные типы объектов применения (воздушная, наземная и морская техника), но и их способностью в перспективе обеспечить повышенный уровень надежности при одновременном сокращении затрат на разработку, производство и сопровождение. Различные производители КБО ИМА, предлагающие свои продукты на рынке, по-разному подходят к реализации основных положений проектирования интегрированных комплексов.

1.1 Аналитический обзор и анализ архитектуры существующих КБО ИМА

Первые разработки интегрированных КБО начались еще в начале 90-х годов XX века в США и были ориентированы на различные классы ЛА. Одним из первых проектов ИМА по праву можно считать систему Airplane Information Management System (AIMS) разработанную для магистрального пассажирского самолета Boeing 777 и сертифицированную для его использования в 1996 г. Другим ранним проектом была система Versatile Integrated Avionics VIA 2000, которая представляла собой распределенную по отдельным блокам модификацию системы AIMS для самолетов старых типов, в том числе военно-транспортных и специального назначения (танкер KC-10, военно-транспортный самолет C-5M, самолет-командный пункт ВМС США E-6) [46]. Однако указанные системы создавались без наличия конкретной методологии проектирования и многих современных стандартов, касающихся ИМА, которые появились лишь в начале 2000-х годов (например, DO-254, ARINC 653).

Основные особенности ИМА можно рассмотреть на примере системы **AIMS**.

Система состоит из двух крейтов, которые соединены с другими системами каналами ARINC 429 и ARINC 629. Крейты выполняют следующие функции [47]:

- управление полетом;
- индикация;
- централизованное техническое обслуживание;
- контроль системы кондиционирования;
- функции связи (в том числе и внутри борта);
- преобразование бортовых интерфейсов (ARINC 429 в ARINC 629).

Крейт системы состоит из 4-х вычислительных модулей (ВМ) и 4-х модулей ввода-вывода (МВВ). Также имеется три свободных слота расширения для установки одного ВМ и двух МВВ (см. рис. 1.1). Таким образом, общими ресурсами платформы являются:

- общие процессоры и единое конструктивное исполнение в корпусе;
- общие порты ввода-вывода и электропитание;
- общая шина данных (SAFEbusTM) для передачи информации между ВМ и МВВ;
- общая операционная система (ОС) с функцией встроенного контроля (built-in test – ВИТ).

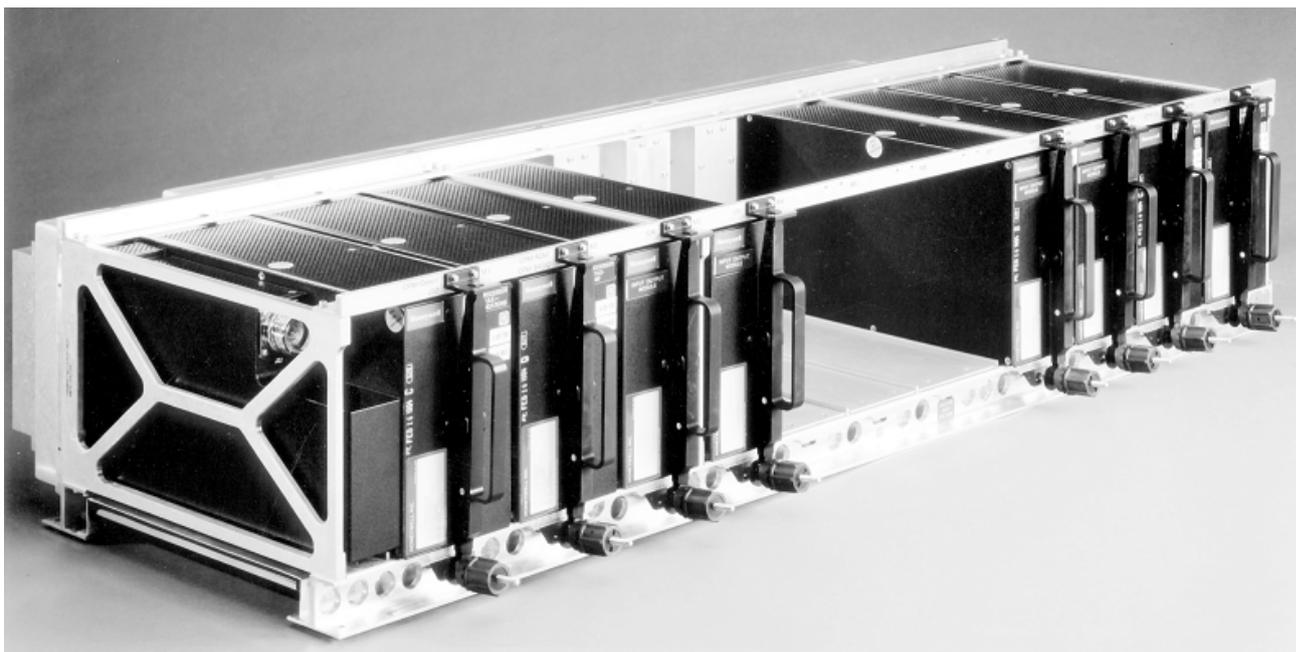


Рисунок 1.1 – Крейт системы AIMS [47]

На аппаратной платформе под управлением ОС запущены следующие программные приложения (с учетом резервных копий):

- программа управления индикаторами (4 шт.);
- программа управления полетом (2 шт.);
- центральное техническое обслуживание (2 шт.);
- программа управления связью (2 шт.);
- программа управления внутрибортовой связью (2 шт.);
- программа контроля кондиционирования воздуха (1 шт.);
- программа сбора полетных данных (2 шт.);
- программа преобразования интерфейсов (4 шт.).

ВМ имеют общую архитектуру с идентичными процессорами, памятью, источниками питания и шиной, но также поддерживают возможность подключения специальной карты ввода-вывода, которая может быть необходима для работы какого-либо приложения.

Система в целом также включает в себя шесть панелей управления, 3 индикатора, устройства управления курсорами и датчики освещенности.

Устойчивость к отказам достигается жесткой изоляцией приложений, дублированием последовательной шины и стопроцентным выявлением возникших сбоев. Таким образом, **AIMS** предоставляет высокую степень функциональной интеграции и сводит к минимуму дублирование вычислительных ресурсов [47].

Другими наиболее известными представителями интегрированных комплексов, запущенных в серийное производство, являются КБО для самолетов Airbus A380 и Lockheed/Boeing F-22 Raptor.

Поставки системы **A380/A400M Integrated modular avionics** начались в 2003 г. Она предназначена для выполнения функций сигнализации, формирования информации для системы индикации (система индикации не входит в состав КБО ИМА), управления радиосвязью, распределения электроэнергии и управления общесамолетным оборудованием. Внешний вид вычислителя и модулей представлен на рисунке 1.2.

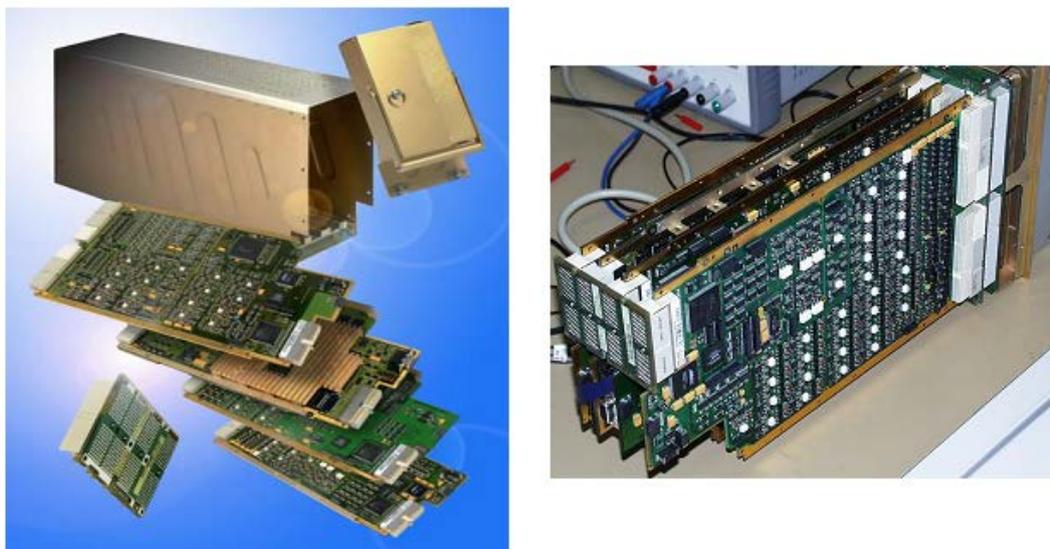


Рисунок 1.2 – Вычислитель и модуль формата 3U самолета A380 [46]

Всего на борту около 30 таких модулей, 18 из которых поставляются фирмой Thales и служат для управления общесамолетным оборудованием [46]:

- 4 для управления шасси;
- 4 для управления топливной системой;
- 4 для управления салоном;
- 4 для управления кондиционированием;
- 2 для управления электроснабжением.

Еще 13 модулей, поставляемых фирмой Airbus Avionics and Simulation, являются модулями общего назначения: 5 VM и 8 MBV.

Связь между модулями осуществляется с помощью высокоскоростного интерфейса AFDX, а управляет всем комплексом ОСПВ, соответствующая стандарту ARINC 653 и разработанная фирмой Thales.

Данный комплекс построен по принципу открытой архитектуры и поддерживает изоляцию программных приложений.

Особым образом стоит выделить систему **Integrated Avionics System (IAS)** самолета F-22 Raptor, архитектура которой представлена на рисунке 1.3. По сведениям разработчиков, эта система является первой полностью интегрированной системой авионики, которая применяется в боевых самолетах США [47].

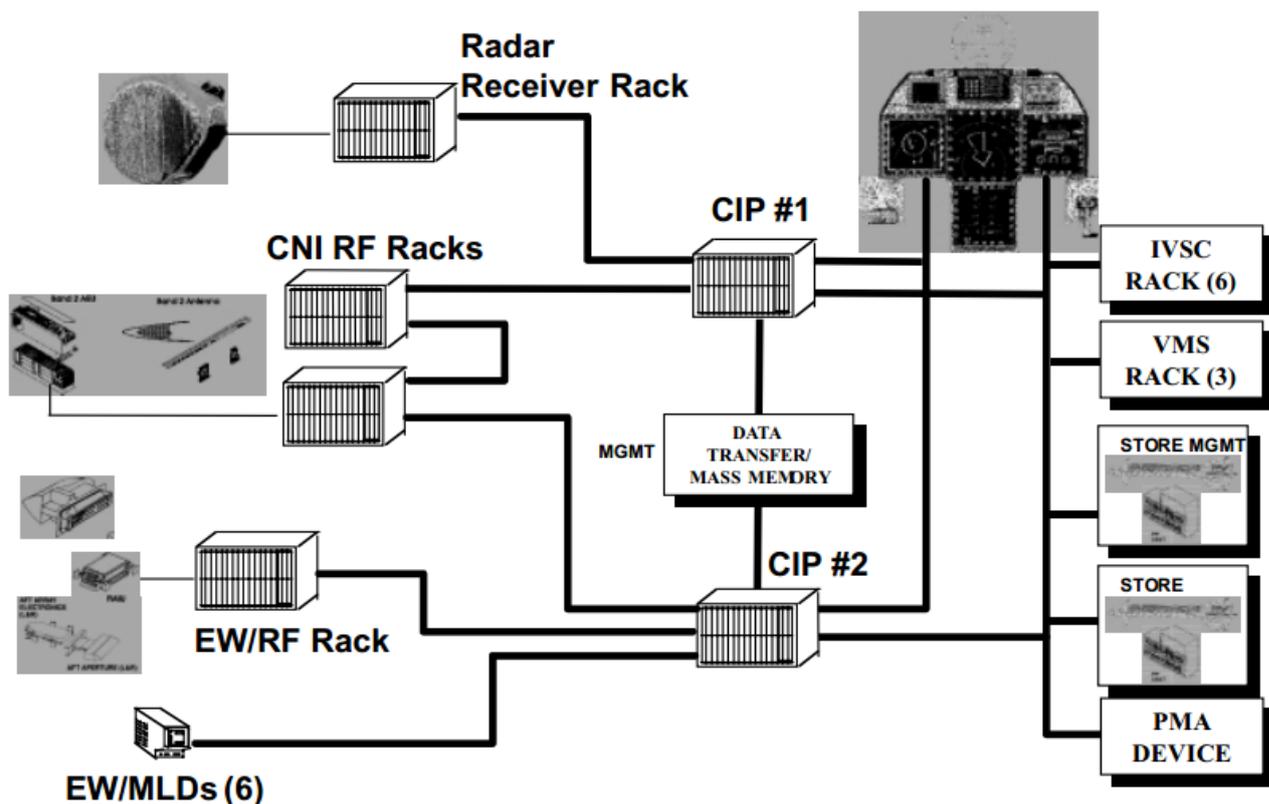


Рисунок 1.3 – Архитектура КБО IAS самолета F-22 Raptor [47]

Комплекс осуществляет обработку информации от датчиков самолета (радар, датчики давления и д.р.), управляет системами электронного противодействия, аппаратурой связи и навигации, а также аппаратурой идентификации.

Ключевыми подсистемами комплекса являются:

- центральный процессор (состоит из двух интегрированных процессоров общего назначения, Common Integrated Processor - CIP);
- подсистема радиоэлектронной борьбы (РЭБ) Electronic Warfare (EW);
- радар;
- подсистема коммуникации/навигации/идентификации Communication/Navigation/Identification (CNI);
- инерциальная система Inertial Reference System (IRS);
- подсистема управления хранилищами данных Stores Management System (SMS);
- элементы управления и индикаторы Controls and Displays (C&D).

Функции крейтов выполняют СІР, каждый из которых имеет жидкостное охлаждение и вмещает по 66 модулей, однако 41 слот зарезервирован под будущее расширение. Также на борту есть место для установки третьего СІР.

Управление осуществляется ОС Avionics Operating System (AOS) и управляющей программой Avionics System Manager (ASM), которые обеспечивают изоляцию приложений друг от друга. AOS предоставляет сервисы, позволяющие нескольким программам разделять ресурсы одного ВМ (Data Processing ment - DPE). Копия AOS и копия ASM выполняются каждым из процессоров DPE.

Ключевой особенностью данного комплекса является обеспечение отказоустойчивости и восстанавливаемости средствами реконфигурации, которая достигается переназначением и/или перепрограммированием модульных ресурсов. Восстановление полного функционала или функционала только выбранного режима полета зависит от количества оставшихся модулей, поддерживающих реконфигурацию. При этом реконфигурация делится на два типа:

- *вторичная реконфигурация* (minor reconfiguration) – осуществляется при потере одного или нескольких модулей в результате отказов;

- *первичная реконфигурация* (major reconfiguration) – осуществляется при отказе целого крейта (например, в результате механических повреждений в бою).

Анализ информации об аппаратной и программной структурах комплекса показывает, что, скорее всего, реконфигурация осуществляется посредством ОС, которая запускает необходимые приложения на свободных функционирующих процессорных модулях, если какой-либо задействованный модуль отказал (об этом говорит большое количество незанятых слотов). Однако не известно, есть ли возможность назначения приложений на уже занятые процессорные модули (т.е. без задействования дополнительных аппаратных ресурсов) путем исключения низкоприоритетных задач при помощи встроенной таблицы приоритетов (prioritization table).

Таким образом, анализ существующих комплексов показывает, что под ИМА понимается концепция построения КБО, базирующая на открытой сетевой архитектуре и единой вычислительной аппаратной платформе [8]. Система, построенная на основе ИМА, обладает набором характерных свойств [46]:

- общая аппаратная платформа;
- модульный состав;
- многозадачность;
- изоляция программных приложений друг от друга;
- централизованное размещение модулей;
- высокоскоростная сеть передачи данных;
- открытая архитектура;
- изоляция программного обеспечения от аппаратной платформы;
- возможность замены модулей в условиях эксплуатации;
- возможность подключения внешних интерфейсов к модулям.

Для постановки задачи исследования необходимо рассмотреть в отдельности те элементы ИМА, на основе которых предполагается реализовать механизмы динамического изменения структуры комплекса. Это аппаратная платформа, функциональные приложения и операционная система реального времени (ОСРВ).

1.1.1 Аппаратная платформа ИМА

Вычислительная платформа ИМА реализуется в виде базовой конструкции (крейта, cabinet) с набором сменных электронных модулей, которые строятся на основе единого конструктивного решения, обеспечивающего принцип унификации и взаимозаменяемости. При этом крейт может содержать и специализированные модули (например, модуль графической обработки, модуль-концентратор и т.д.).

Аппаратная часть комплекса в общем случае делится на три уровня [18, 48-51]:

- *первый уровень* образуют унифицированные конструктивно-функциональные модули (КФМ), выполненные в едином конструктивном испол-

нении (например, по европейскому стандарту типоразмеров 3U, 6U или 9U), имеющие одинаковый набор аппаратных ресурсов и выполняющие разные функции.

В состав таких модулей также могут входить дополнительные функциональные узлы, так называемые *мезонины*, которые устанавливаются с помощью специального соединителя, предусмотренного в конструкции соответствующего КФМ;

- *второй уровень* образуют многопроцессорные вычислительные системы, выполненные в виде стандартного корпуса (крейта), которые реализованы на основе модулей первого уровня;

- *третий уровень*, который представляет собой высокоскоростную бортовую сеть передачи данных на основе какого-либо сетевого интерфейса высокой пропускной способности, служащую интегратором крейтов второго уровня.

Новейшими отечественными платформами, построенными на основе указанной аппаратной архитектуры, являются «Платформа интегральной модульной авионики Крейт 6U 10М» производства ФГУП «СПб ОКБ «Электроавтоматика» и «Бортовая центральная вычислительная система БЦВС-1» производства ОАО «РПКБ «Раменское».

Крейт 6U 10М является симметричной многопроцессорной системой с дублированной средой коммутации каналов стандарта ARINC 664 (см. рис. 1.4) [52], который состоит из:

- вычислительных/интерфейсных модулей – 10 шт.;
- модулей питания – 2 шт.;
- выходных соединителей на тыльной стороне крейта;
- оптических соединителей – 48 шт.;
- внешних интерфейсы ARINC 664 (Fast Ethernet 10/100 Base-T), ARINC 818 (Fibre Channel 1x, используется многомодовый волоконно-оптический кабель 50/125 мкм, рабочая длина волны 1310 нм).

Габариты крейта 490x280x330 (LxВxН), масса 35 кг.



Рисунок 1.4 – Крейт 6U 10M производства
ФГУП «СПб ОКБ «Электроавтоматика» [52]

В состав крейта входят следующие сменные модули (см. рис. 1.5):

- модуль процессорный общего назначения МПОН-6-001;
- модуль вычислительный МВ80-001;
- модуль носитель мезонинов МНМ-6-001;
- модуль графический МГ-6-001;
- модуль вторичного электропитания МВЭ-6-001-000;
- мезонин интерфейсных каналов по ГОСТ 18977-79 (ARINC 429);
- мезонин интерфейсных каналов ARINC 818 и AFDX.

По мнению производителя, особенностями и преимуществами платформы типоразмера 6U являются:

- возможность достижения более высокого уровня системной интеграции (большое количество контактов для организации внешних связей) по сравнению с формфактором 3U;
- возможность достижения более высокой удельной вычислительной мощности (отношение вычислительной мощности к массогабаритным характеристикам) по сравнению с формфактором 3U;
- возможность размещения электронных компонентов с дополнительными сервисно-диагностическими функциями;
- уменьшение «накладных расходов» на конструкцию.

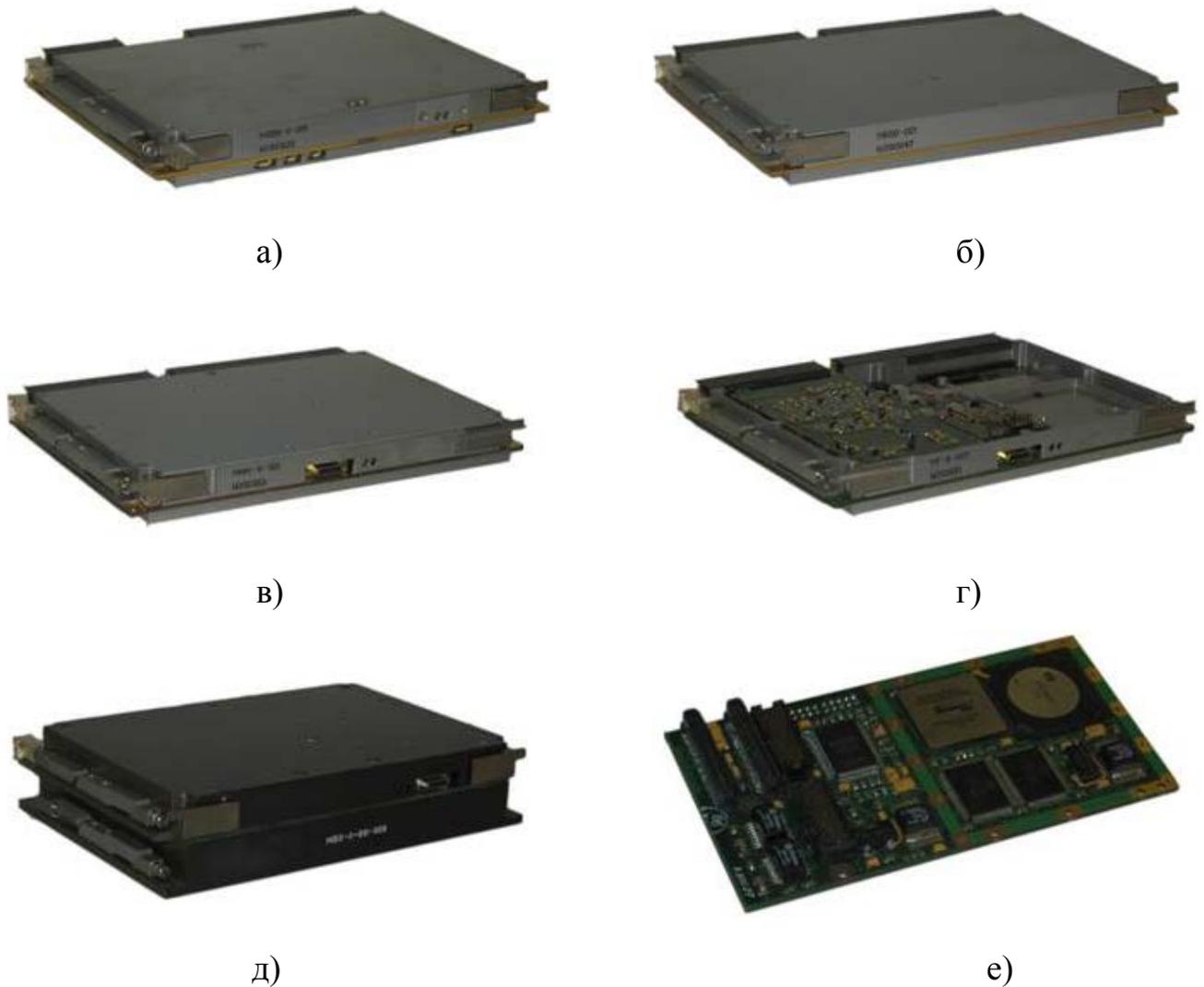


Рисунок 1.5 – КФМ крейта 6U 10M [52]:

а) МПОН-6-001, б) MB80-001, в) МНМ-6-001, г) МГ-6-001

д) MBЭ-6-001-000, е) Мезонин ARINC 818 и AFDX

Дополнительным преимуществом конструкции приведенных модулей является устойчивость к специальным внешним воздействующим факторам (ВВФ), что позволяет использовать их в военной технике. На данный момент завершена автономная настройка крейта и производится комплексная отработка в составе демонстрационного стенда ФГУП «ГосНИИАС».

Платформа БЦВС-1 также является симметричной многопроцессорной системой с дублированной средой коммутации каналов связи, но, в отличие от крейта 6U 10M, строится на основе модулей типоразмера 3U и предназначена для выполнения вычислительных и управляющих функций в составе информационно-

управляющих систем перспективных комплексов бортового оборудования военной и гражданской авиации [53]. Внешний вид платформы представлен на рисунке 1.6.



Рисунок 1.6 – Внешний вид БЦВС-1 [53]

В состав крейта входят следующие модули (см. рис. 1.7):

- модуль процессора общего назначения МПОН-2;
- модуль коммутации МК-15М;
- модуль вторичного электропитания МВЭ-1;
- модуль расширения авиационные интерфейсов МРАИ;
- вычислительно-интерфейсный модуль общего назначения ВИМ-3U-2;
- модуль сетевого коммутатора МСК-3U-2.

Габаритные размеры крейта соответствуют формфактору 5,5К по ГОСТ 23701-79, масса составляет 15 кг, напряжение питания от сети постоянного тока напряжением 27 В по ГОСТ 19705-89.

Габаритные размеры и масса крейта БЦВС-1 практически в 3 раза меньше по сравнению с 6U 10M, что делает возможным его применение на самолетах малой авиации или в качестве дополнительного комплекса при наличии других систем, например традиционной федеративной архитектуры.

Однако крейтовое исполнение не является обязательным качеством платформ ИМА, более того, это качество практически никогда не реализовывается до конца, что проявляется в наличии хотя бы двух крейтов. Это обусловлено обеспечением дополнительной надежности и удобством размещения оборудования на борту ЛА.



а)



б)



в)



г)



д)



е)

Рисунок 1.7 – КФМ крейта БЦВС-1 [53]:
а) МПОН-2, б) МК-15М, в) МВЭ-1, г) МРАИ
д) ВИМ-3У-2, е) МСК-3У-2

Встречается также более гибкое решение, которое предполагает одновременное расположение модулей ИМА в крейтах различного типа, например стандартных типоразмеров (3U, 6U, 9U) и корпусах многофункциональных индикаторов. Это решение наиболее эффективно при использовании систем на бортах малых ЛА, которые предъявляют повышенные требования к массогабаритным характеристикам комплексов. Такую архитектуру называют *распределенной ИМА* [46].

КБО ИМА также должен поддерживать возможность связи с другими системами, не входящими в его состав, например по КЛС стандарта ARINC 429. Для этого, как правило, предусмотрен специальный модуль, на котором реализованы интерфейсы сопрягаемых систем.

Анализ аппаратуры представленных комплексов показал, что она строится на основе общедоступных COTS (Commercial Off The Shelf) компонентов, которые в состоянии удовлетворить растущую потребность в высокопроизводительной элементной базе. Так, например, емкость программируемых логических интегральных схем (ПЛИС) уже сейчас достигает нескольких сотен тысяч логических ячеек. Такие ресурсы позволяют в перспективе осуществлять перестроение структуры комплексов любой сложности.

Однако для техники специального назначения необходима элементная база, стойкая к различным ВВФ. Для этого разработаны специализированные электро-радиоизделия (ЭРИ), которые по своим характеристикам в десятки раз уступают коммерческим и обладают целым рядом ограничений (например, радиационно-стойкие ПЛИС не имеют возможности многократной перезаписи). Для удовлетворения требований стандартов на воздействие специальных ВВФ в комплексах подобного рода используются технологические и конструкторские методы, позволяющие успешно пройти периодические или приемосдаточные испытания даже при использовании компонентов, не рассчитанных на столь широкий диапазон воздействующих факторов.

1.1.2 Функциональные особенности ИМА

Ключевой особенностью функционирования КБО ИМА является наличие программных приложений разного уровня критичности, которые выполняют заданные функции и разделяют общие аппаратные ресурсы. При этом каждому приложению необходим вычислитель и некоторый объем памяти для хранения промежуточных результатов вычислений.

Поскольку крейт ИМА содержит в себе набор базовых ВМ, на каждом из которых выполняется какое-либо функциональное приложение (или несколько приложений), то он является многопроцессорной системой (МС). Тип МС влияет на распределение памяти в системе и количество одновременно запущенных копий ОСРВ, что непосредственно влияет на выбор механизмов реконфигурации.

В соответствии с [16] по способу организации пространства памяти многопроцессорные системы делятся на системы с общей памятью и системы с передачей сообщений (с локальной памятью). Их структура представлена на рисунке 1.8 (глобальные распределенные системы не рассматриваются в виду их неприменимости по отношению к КБО).

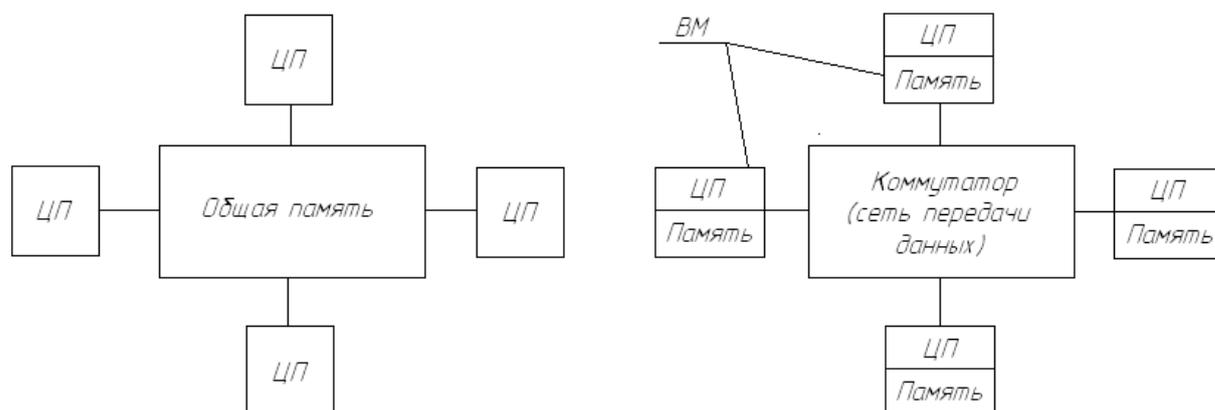


Рисунок 1.8 – Типы многопроцессорных систем:

а) система с общей памятью, б) система с передачей сообщений

В МС типа а) обмен данными идет через общую память. В этой системе каждый процессор имеет равный доступ ко всей физической памяти. Доступ к слову памяти обычно занимает 2-10 нс. При всей своей кажущейся простоте данный тип

системы обычно включает в себя передачу большого количества защищенных сообщений.

В системе типа б) каждый модуль памяти является локальным по отношению к одному центральному процессору. Процессоры связываются друг с другом путем отправки сообщений по сети передачи данных. При наличии высокоскоростных каналов связи сообщения могут быть отправлены за 10-50 мс, что намного превышает время доступа к памяти в системе а). Системы с передачей сообщений гораздо проще реализовать, чем системы с общей памятью, но они труднее в программировании [16].

Анализ серийно выпускаемых вычислительных модулей формата 6U показывает, что все они имеют свою локальную память. Поэтому известные крейты ИМА соответствуют типу б) многопроцессорной системы. Поскольку задержки передачи данных в такой системе намного больше, то проектировать ее нужно с учетом минимизации сетевого трафика [16]. Для этого необходимо распределить программные приложения по определенному оптимальному алгоритму.

При этом в крейте присутствует и модуль глобальной памяти, который осуществляет сбор статистической информации. Поэтому система представляет собой комбинацию видов а) и б), изображенных на рисунке 1.3.

Поскольку управление приложениями осуществляется исключительно ОСРВ, то производить реконфигурацию программной части в случае отказов возможно только встроенными в нее средствами, пусть даже с использованием каких-либо сервисных приложений. Поэтому тип организации ОСРВ будет оказывать решающее влияние на методы и средства реконфигурации. Известны три типа организации ОСРВ в многопроцессорной системе ИМА.

Простейшим вариантом реализации является запуск своей собственной ОСРВ на каждом из ВМ крейта (см. рис. 1.9).

Вполне очевидна оптимизация, позволяющая всем ВМ совместно использовать код операционной системы и создавать собственные копии только лишь структуры данных ОСРВ.

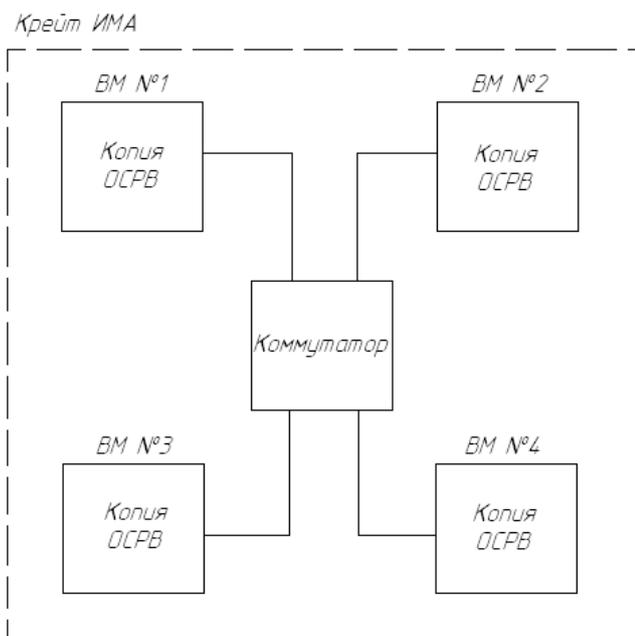


Рисунок 1.9 – Четыре VM, использующие общий код ОСРВ

Вторая модель, модель master-slave, приведена на рисунке 1.10. Здесь используется лишь одна копия операционной системы, а ее таблицы существуют исключительно для одного VM. Все системные вызовы переадресовываются ему для дальнейшей обработки.

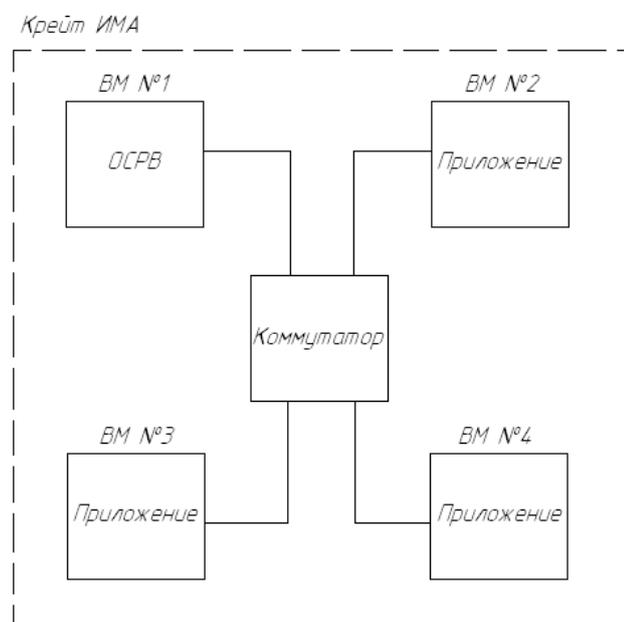


Рисунок 1.10 – Крейт, построенный по модели “master-slave”

В этой модели, в отличие от первой, очень хорошо организовано распределение приложений между VM. Когда один VM остается без работы или выполняемая им низкоприоритетная задача исключается из состава комплекса, он посылает

запрос операционной системе ВМ №1 на предоставление ему приложения, готового к работе и требующего аппаратных ресурсов.

Проблема данной модели заключается в том, что при достаточно большом количестве ВМ главный процессор становится ее узким местом. Если, например, 10% всего времени он тратит на обработку системных вызовов, то 10 ВМ дадут ему предельную нагрузку. Таким образом, эта модель подходит только для небольших многопроцессорных систем [16]. К тому же при изменении структуры КБО ИМА решение о назначении того или иного приложения на ВМ будет приниматься вычислителем, отвечающим за реконфигурацию и производящим мониторинг состояния всей системы. Поэтому запросы на назначение задач от других ВМ не нужны.

Асимметрия модели master-slave устраняется в симметричных мультипроцессорах (SMP). В памяти присутствует только одна копия операционной системы, но ее может запустить любой ВМ (см. рис. 1.11).

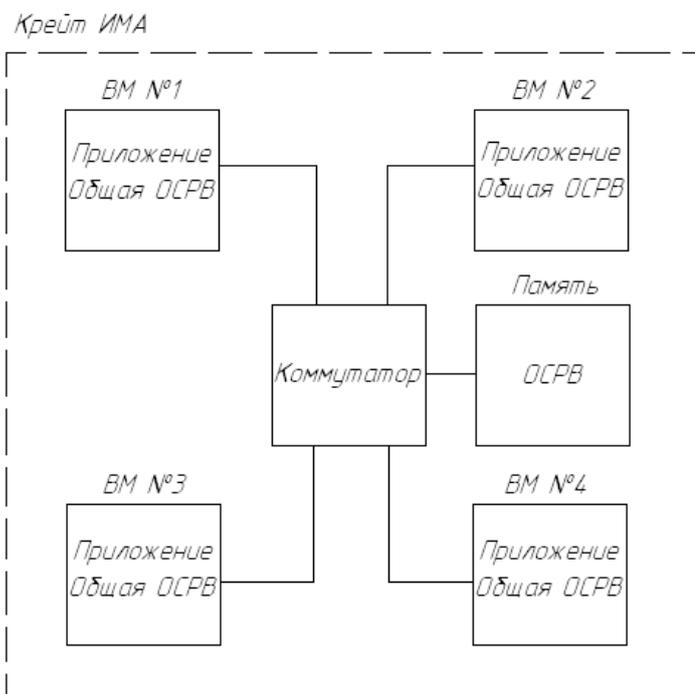


Рисунок 1.11 – Крейт, построенный на основе модели SMP

Однако если в данной системе два или более ВМ запускают код операционной системы одновременно (например, при перераспределении приложений по аппаратной платформе во время реконфигурации), то может произойти взаимоблоки-

ровка. Простейший способ обхода этой проблемы – связывание с каждой ОС мьютекса (то есть блокировки). Если мьютекс заблокирован, то ВМ просто переходит в режим ожидания. Но данное решение неприменимо для КБО с ОСРВ, поскольку за время ожидания информация может потерять актуальность. К тому же предсказать поведение такой системы будет гораздо сложнее.

С точки зрения реализации реконфигурации наиболее приемлемым выглядит комбинированный подход, объединяющий модель master-slave с моделью, предполагающей наличие ОСРВ на каждом вычислительном модуле крейта [54, 55]. Это объясняется двумя причинами:

- поскольку каждый ВМ обладает своей собственной памятью, то удобнее всего превратить его в автономный вычислитель. Это минимизирует потери от его отказа, в отличие от ситуации, если бы он был, например, носителем общей ОСРВ;

- наличие дополнительного модуля – носителя ОСРВ необходимо для поддержания возможности реконфигурации. Он должен следить за состоянием остальных модулей и назначать на них программные приложения. В случае отказа такого модуля теряется только дополнительная живучесть – остальные модули продолжают работу.

Программные приложения, работающие под управлением ОСРВ, как правило, пишутся на языках высокого уровня и разрабатываются разными производителями ПО. Такие приложения должны поддерживать возможность работы в условиях реального времени и иметь дополнительные механизмы обеспечения безопасности при реализации функций высокого уровня критичности.

1.1.3 Операционная система реального времени

КБО на базе ИМА принято относить к системам реального времени. Согласно определению, данному М. Тиммерманом и Б. Бенеденом, *система реального времени* – это система, которая предсказуемо (в смысле времени реакции) реагирует

на непредсказуемые (по времени появления) внешние события [56]. В общем виде такой комплекс имеет структуру, представленную на рисунке 1.12.

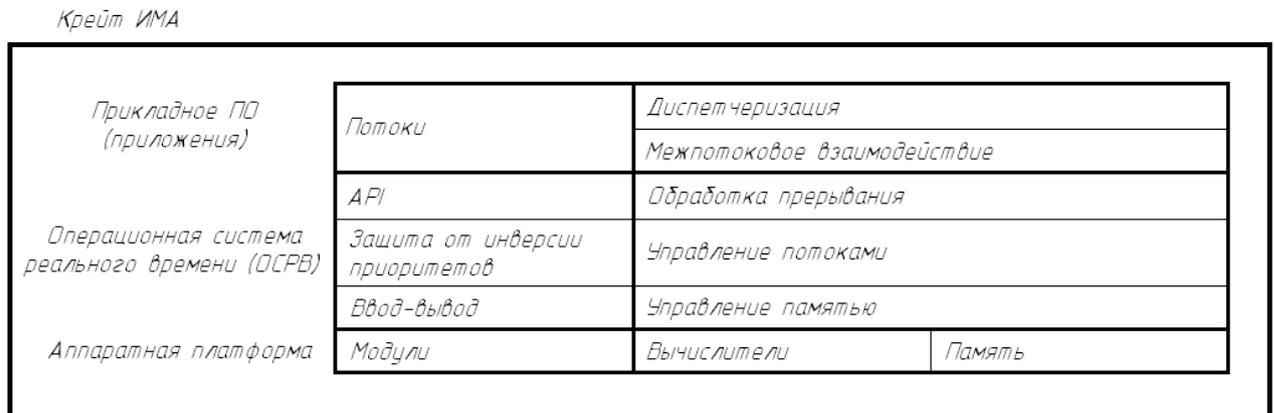


Рисунок 1.12 – Структура системы реального времени

Основным инструментом, обеспечивающим распределение задач в комплексе ИМА и осуществляющим контроль над их исправной работой, является ОСРВ. Поскольку большое количество задач, выполняемых комплексом, имеют самый высокий уровень критичности, то используемая ОСРВ должна относиться к системам жесткого реального времени, которые не допускают возникновения задержек при реакции на какое-либо событие (например, отказ). Такая система использует следующие механизмы, обеспечивающие предсказуемость поведения комплекса:

- *система приоритетов и алгоритмы диспетчеризации.* В отличие от ОС общего назначения, где используется какая-либо модификация алгоритма круговой диспетчеризации, которые используют термин непрерывного кванта времени (“time slice”), предоставляемого приложению для работы, ОСРВ имеет возможность сменить выполняемое приложение до окончания этого кванта;

- *механизмы взаимодействий между приложениями.* К ним относятся средства синхронизации процессов и передачи данных между ними (семафоры, мьютексы, события, сигналы, средства для работы с разделяемой памятью, каналы данных, очереди сообщений). Эти механизмы являются очень важным с точки зрения обеспечения реконфигурации, поскольку именно на их основе сервисные приложения будут осуществлять принятие решений о перераспределении задач;

- средства для работы с таймерами. Эти средства обеспечивают возможность задания различных интервалов времени, генерировать прерывания по их истечении, а также задавать будильники.

Современные ОСРВ строятся на основе различных архитектур, которые обладают своими достоинствами и недостатками, а также подходят для организации систем различного типа.

Монолитные ОС (рис. 1.13) построены таким образом, чтобы предоставлять программным приложениям необходимые интерфейсы для взаимодействия с аппаратурой через ядро системы. Главным недостатком такой архитектуры является плохая предсказуемость, обусловленная сложным взаимодействием модулей системы друг с другом.

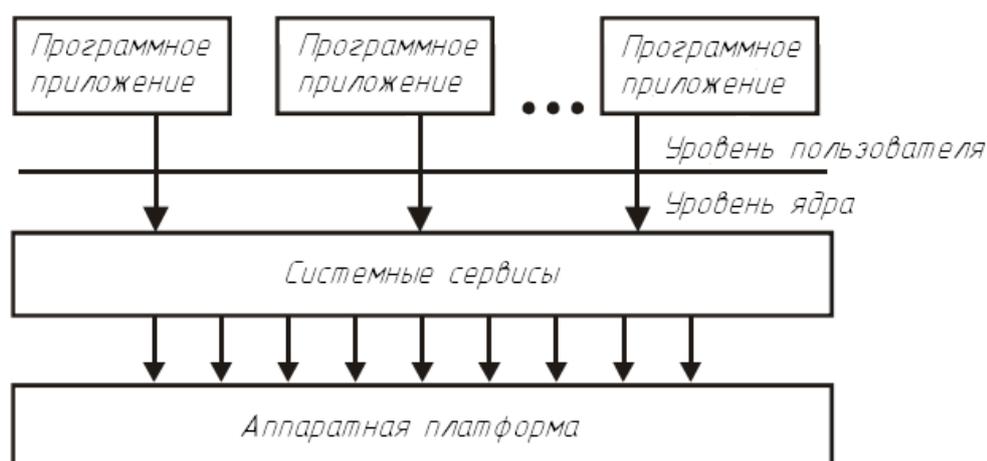


Рисунок 1.13 – Структура монолитной ОС

В *уровневых ОС* (рис. 1.14) программные приложения могут получить доступ к аппаратным ресурсам не только через ядро системы или ее специализированные сервисы, но и непосредственно. Такая система обеспечивает большую предсказуемость реакций, а также позволяет приложениям быстрее получать доступ к необходимым ему аппаратным средствам. Одним из основных недостатков данного вида ОС является отсутствие многозадачности, которая требуется для реализации КБО ИМА и, в частности, алгоритмов реконфигурации.

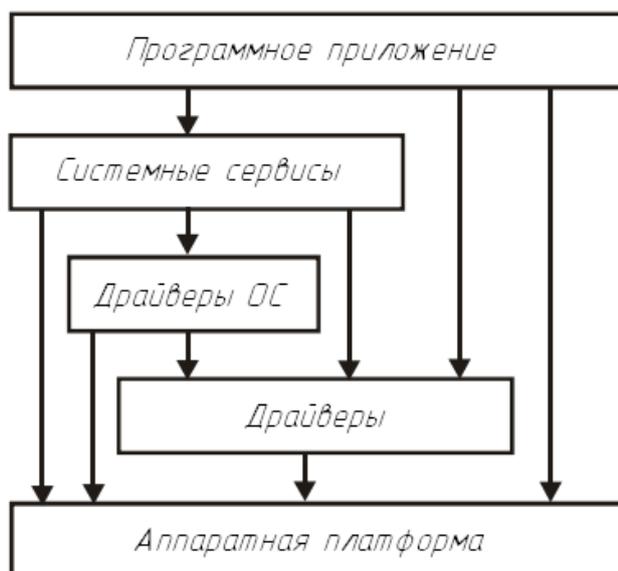


Рисунок 1.14 – Структура уровневой ОС

Наиболее перспективной архитектурой ОС, с точки зрения построения реконфигурирующегося КБО ИМА, является *микроядерная архитектура* (рис. 1.15).

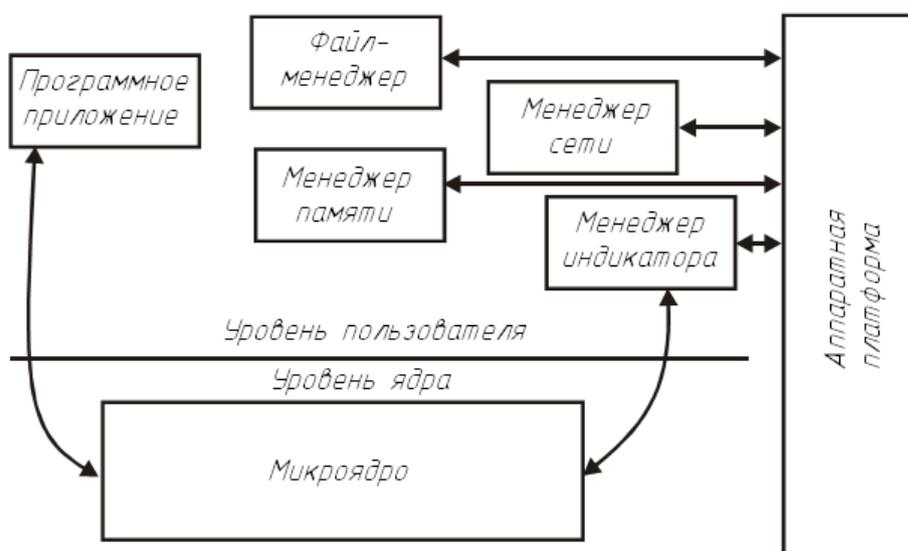


Рисунок 1.15 – Структура микроядерной ОС

На сегодняшний день на рынке представлены несколько коммерческих ОСРВ авиационного применения (см. таблицу 1.1). Большая часть из них построена на именно основе микроядерной архитектуры или архитектуры “клиент-сервер”, которая фактически представляет собой микроядро с двумя обособленными классами процессоров: серверами, предоставляющими те или иные службы, и клиента-

ми, использующими эти службы [16]. Объясняется это толерантностью таких систем к отказам на уровне ядра, то есть ошибка в одном из компонентов приведет к завершению процесса компонента, а не к отказу всей ОС.

Микроядерные ОСРВ обладают целым рядом преимуществ перед монолитными с точки зрения проектирования многопроцессорных реконфигурирующихся комплексов ИМА:

- простота реализации и малые размеры исполняемого кода системы из-за четко определенной функциональности ядра, что позволяет запускать копии системы на каждом модуле крейта (в соответствии с принципом, описанным в п. 1.1.2);

- более высокая надежность за счет управления отдельными функциональными приложениями, в которых легче отследить ошибки;

- улучшенные возможности масштабирования за счет исключения отказавших приложений из комплекса без ущерба для его дальнейшего функционирования;

- повышенная отказоустойчивость за счет повторного запуска приложения (в том числе и отказавшего) без перезагрузки всей системы.

Главным недостатком микроядерной ОСРВ является более низкая производительность из-за затрат ресурсов на межпроцессорное взаимодействие.

Очень хорошо микроядерные ОСРВ показывают себя с точки зрения временных параметров, а именно [57]:

- времени отклика на прерывание;
- времени переключения между двумя потоками в одном процессе;
- времени переключения контекста процесса, то есть между двумя потоками управления, принадлежащими двум различным процессам.

Таблица 1.1 – ОСРВ коммерческого применения

Свойства	ОСРВ			
	QNX Neutrino	WxWorks	Integrity-178B	LynxOS
Архитектура	Микроядерная, клиент-сервер	Микроядерная, клиент-сервер	Микроядерная	Монолитное ядро
Поддержка многопроцессорных систем	да (QNX MultiCore Technology)	да (только асимметричные системы, WxMP)	нет	да (SMP, не сертифицирована для авиации)
Толерантность к отказам	да (QNX High Availability Toolkit)	нет	да	да (только на уровне приложений)
Поддержка распределенных систем	да (QNX Transparent Distributed Networking)	да (только с использованием WindNet)	-	да
Механизмы мониторинга	да (QNX System Analysis Toolkit)	да (Wind Scheduling)	да (диспетчеризация и контроль)	да (SpyKer)
Изоляция приложений	да (QNX Adaptive Partitioning, аналог ARINC 653)	нет (кроме установки WxMVI)	да (ARINC 653)	да (MMU, виртуальная память)
Поддержка стандарта POSIX	да (POSIX 1003.1-2001)	да (POSIX 1003.1, .1b, .1c)	да	да (POSIX 1003.1a/b/c, 1003.1-2003)
Поддерживаемые процессорные архитектуры	ARM, MIPS, PowerPC, SH-4, X86	ARM, Intel, PowerPC	ARM, MIPS, PowerPC, Blackfin, X86	Freescale, Intel
Открытость исходного кода	да (по отдельной лицензии)	да (по отдельной лицензии)	нет	да (по отдельной лицензии)
Качество документации тех. поддержки	высокое	среднее	высокое	среднее
Доступность на российском рынке	да	да	нет	да

Микроядерные ОСРВ обладают целым рядом преимуществ перед монолитными с точки зрения проектирования многопроцессорных реконфигурирующихся комплексов ИМА:

- простота реализации и малые размеры исполняемого кода системы из-за четко определенной функциональности ядра, что позволяет запускать копии системы на каждом модуле крейта (в соответствии с принципом, описанным в п. 1.1.2);

- более высокая надежность за счет управления отдельными функциональными приложениями, в которых легче отследить ошибки;

- улучшенные возможности масштабирования за счет исключения отказавших приложений из комплекса без ущерба для его дальнейшего функционирования;

- повышенная отказоустойчивость за счет повторного запуска приложения (в том числе и отказавшего) без перезагрузки всей системы.

Главным недостатком микроядерной ОСРВ является более низкая производительность из-за затрат ресурсов на межпроцессорное взаимодействие.

Очень хорошо микроядерные ОСРВ показывают себя с точки зрения временных параметров, а именно [57]:

- времени отклика на прерывание;
- времени переключения между двумя потоками в одном процессе;
- времени переключения контекста процесса, то есть между двумя потоками управления, принадлежащими двум различным процессам.

Результаты сравнения временных характеристик ОСРВ QNX Neutrino, VxWorks AE-653 и LynxOS приведены в таблице 1.2. Временные характеристики микроядерных систем во несколько раз превосходят аналогичные для монолитной LynxOS, что позволяет сделать вывод о предпочтительности использования клиент-серверных ОСРВ в перспективном реконфигурирующемся комплексе, поскольку в нем необходимо максимально быстрое перераспределение задач при возникновении отказов.

Таблица 1.2 Сравнение временных параметров ОСРВ

Тест	QNX Neutrino		VxWorks AE-653		LynxOS [57]	
	Сред.	Макс.	Сред.	Макс.	Сред.	Макс.
Время отклика на прерывание, мкс	1,7	4,1	1,7	6,8	3	11
Время переключения между 2 потоками, мкс	2,0	8,1	2,9	15,5	11	41
Время переключения между 10 потоками, мкс	2,4	7,4	3,4	16,5	-	-
Время переключения между 128 потоками, мкс	7,2	15,9	6,8	46,8	-	-

Однако следует понимать, что данные результаты получены при тестировании ОСРВ на различном оборудовании, при этом состав и тип тестов остается неизвестным.

Кроме абсолютных значений временных параметров ОСРВ очень важной является и их стабильность, которая и определяет детерминированность времени обработки данных и момента выдачи результатов.

Исходя из данных журнала Dedicated System, QNX Neutrino опережает по этому критерию VxWorks AE-653 как по разбросу характеристик в серии однотипных тестов (отношение максимума времени к среднему значению существенно меньше), так и с ростом нагрузки (время переключения потока при увеличении числа потоков от 2 до 128 у QNX Neutrino выросло только в 1,65 раза, тогда как у VxWorks AE-653 – в 2,24 раза) [57].

Ключевым аспектом при выборе ОСРВ является поддержка ими многопроцессорных систем, что выражается в наличии удобных и надежных средств организации межмодульного взаимодействия и обеспечения сетевой прозрачности.

ОСРВ QNX Neutrino предлагает поддержку многопроцессорных систем. Для VxWorks AE-653 такая поддержка пока анонсирована только для асимметричных систем при установке дополнительного пакета WxMP. Авиационной версии LynxOS с поддержкой многопроцессорных плат пока тоже не существует.

В отношении сетевых протоколов нужно отметить, что ОСРВ LynxOS, VxWorks AE-653 и QNX Neutrino обладают примерно равными (и широкими) возможностями. Дополнительным плюсом QNX Neutrino является ее специальная архитектура сетевой подсистемы, обеспечивающая сетевую «прозрачность» прикладных программ: любой процесс может вызвать другой процесс на удаленном модуле так же, как и процесс на локальном модуле [57].

1.2 Современные стандарты проектирования КБО ИМА

Основным нормативным документом, определяющим цели, процессы и мероприятия для разработчиков КБО ИМА, является «Руководство Р-297 по вопросам разработки и квалификации интегрированной модульной авионики» [58], введенное в действие Авиационным Регистром Межгосударственного Авиационного Комитета (АР МАК) 20 марта 2015 года директивным письмом № 05-2015.

В данном документе, помимо описания концепции ИМА и рекомендаций по квалификации (сертификации и валидации) соответствующих комплексов, содержатся общие требования и требования по распределению функций, обеспечивающему готовность, целостность и безопасность проектируемой системы, а именно:

1. Требования к готовности. Распределение функций ЛА с учетом архитектуры ИМА, распределение, управление и защита ресурсов платформы ИМА, надежность и контроль рабочего состояния.
2. Требования к целостности. Гарантия проектирования, обнаружение неисправностей и их обособление.
3. Требования безопасности. Защита от отказов, оценка летной годности.
4. Требования к управлению неисправностями. Идентификация отказов и соответствующие действия по их устранению.
5. Требования расширяемости. Возможность интеграции нового приложения без нарушения требований уже интегрированного приложения. Разделение приложений (робастное обособление).

Чтобы обеспечить эти требования, при разработке платформы ИМА должны быть выполнены следующие действия:

1. Определен полный перечень функций ЛА;
2. Определен перечень функций, назначаемых на платформу ИМА, с указанием требуемых технических характеристик.
3. Разработана платформа ИМА с учетом, в том числе, следующих аспектов:
 - количества, качества и типов модулей и ресурсов платформы ИМА;
 - требований к функциям ЛА, например, необходимости размещения на определенном типе модуля;
 - требований к целостности при обеспечении разнородности вычислений (сравнение результатов вычислений для достижения необходимой надежности).
4. Проведена оценка воздействия на ЛА комбинации отказов платформы, поддерживаемых приложений и совместно используемых ресурсов.
5. Определены изменения, необходимые для назначения ресурсов платформы ИМА с целью разрешения проблем, обнаруженных в ходе отдельных оценок.
6. Реализована система ИМА следующим образом:
 - разработаны приложения и проведена их верификация;
 - проведена интеграция программной и аппаратной частей системы ИМА;
 - разработана методика первичного анализа отказов системы ИМА на основе событий верхнего уровня;
 - проведена оценка комбинации отказов компонентов платформы ИМА, влияющих на выполнение ей своих функций, и осуществлена соответствующая корректировка назначения и/или применения приложений;
 - проведены наземные и летные испытания ЛА на соответствие оценке безопасности системы.
7. Проведена интеграция, валидация и верификация комплекса и получены соответствующие документы, подтверждающие соответствие системы предъявляемым требованиям.

Отдельное внимание уделяется назначению ресурсов системы ИМА. Отмечается, что в дополнение к специфическим функциям ЛА могут потребоваться и специальные ресурсы приложения ввода/выводы данных, а также соответствующие аппаратные модули. В связи с этим к платформе ИМА предъявляются дополнительные требования [58]:

- вычислительная производительность системы ИМА должна соотноситься с требуемым временем выполнения приложений, временем переключения контекстов, накладными затратами платформы и другими требованиями обработки данных;

- характеристики аппаратной части и пропускная способность сети должны учитывать затраты (включая разделение приложений, обработку данных, нестабильность шины данных и информационный трафик), связанные с переключением между процессами, распределением данных и другими ограничениями;

- для удовлетворения требований по безопасности ЛА платформа ИМА и поддерживаемые приложения определяют требования целостности и готовности назначенных функций и конфигурацию системы ИМА.

Руководство Р-297 тесно связано с другими национальными стандартами, определяющими правила разработки ПО (КТ-178В [59]) и аппаратуры (КТ-254 [60]), обеспечения безопасности авионики (Р-4761 [61] и Р-4754 [62]). Однако эти документы затрагивают лишь процесс непосредственной реализации комплекса ИМА и оценки его безопасности, но при этом не дают рекомендаций или правил по интеграции ПО и аппаратуры, по проведению оценки воздействия на ЛА комбинации отказов платформы и разработке соответствующих решений по их парированию.

Таким образом, с точки зрения соответствия требованиям новейшего стандарта на проектирование КБО ИМА, становятся актуальными следующие задачи:

1. Разработка моделей комплекса с учетом специфики концепции ИМА и требований стандартов на проектирование и обеспечение безопасности систем авионики.

2. Разработка методов распределения ресурсов и первичного анализа отказов системы ИМА на основе событий верхнего уровня с учетом требований стандартов на проектирование ИМА.

3. Разработка методов парирования отказов с использованием доступных программных (ОСРВ и функциональные приложения) и аппаратных (модули и крейты) средств для достижения повышенных показателей надежности.

1.3 Современные методы проектирования КБО ИМА

Общепринятый подход к разработке КБО ИМА заключается в интеграции аппаратуры комплекса в единое вычислительное ядро, которая основывается на объединении бортовых систем на всех этапах разработки, начиная с эскизного проекта и заканчивая реализацией стандартных унифицированных КФМ [63].

Наиболее актуальными в области исследования и проектирования ИМА в настоящее время являются работы Жаринова И.О. [18-21]. Он проанализировал многолетний опыт создания КБО на различных предприятиях и пришел к выводу, что большая часть элементов авионики состоит из готовых наборов алгоритмов обработки информации, взаимозаменяемых КФМ, блоков и подсистем. На основе этих выводов им были разработаны математические модели наиболее значимых сущностей КБО:

- модель состава объекта проектирования, которая представляется в виде матрицы функциональных элементов;
- модель структуры объекта проектирования в виде матрицы соединений функциональных элементов;
- модель функционирования объекта проектирования;
- модель параметров в виде вектора различных показателей (технических, экономических и т.д.).

Проблема проектирования при этом была сформулирована как поиск однозначного отображения множества предъявляемых требований ТЗ в базис реализуемых характеристик комплекса. На основе этого базиса, при помощи генетиче-

ских алгоритмов и алгоритмов генерации проектных решений, были предложены различные варианты реализации комплекса авионики на основе уже существующих. При этом критерием оптимальности выступал минимум отклонения вектора результирующих характеристик от вектора требований ТЗ.

Декомпозиция вычислительной системы на функциональные элементы при этом заключалась в выделении независимых подграфов из многосвязного графа операций обмена и обработки информации с целью их реализации на отдельных вычислителях. Однако такой подход не учитывает специфику организации вычислительно процесса в ОСРВ и типы функциональных задач. Поскольку каждая прикладная функция реализована в виде сложного программного приложения, которое не может выполняться одновременно на двух вычислителях, то нет необходимости в выделении независимых частей из множества функций и операторов (такими частями являются сами приложения). Необходимо также принимать во внимание и структуру крейта ИМА, в котором данные могут передаваться как от одного ВМ к другому напрямую, так и посредством общей памяти МЗУ базы данных. Это зависит, прежде всего, от выбранного типа мультипроцессорной системы и влияет на способ распределения приложений по аппаратной платформе.

При этом интеграция заключается лишь в объединении аппаратной части в единое целое, а программные приложения жестко закрепляются за соответствующими модулями.

Также в работах Жаринова И.О. не рассматриваются критичность выполняемых функций (в соответствии с современными стандартами) и возможность возникновения отказов, в результате чего остаются не полностью раскрытыми такие важные вопросы как динамическое изменение структуры комплекса и оптимальное распределение приложений по аппаратной платформе.

Задача оптимизации структуры вычислительных систем решается учеными уже на протяжении многих десятилетий. ИМА как многопроцессорная система с точки зрения оптимального распределения ресурсов имеет несколько ключевых особенностей:

1. Принадлежность к системам жесткого реального времени. Реакция такой системы на внешние события и воздействующие факторы должна быть предсказуемой и детерминированной.

2. Выполнение системой функций с разным уровнем критичности. Отказ определенной функций, в зависимости от ее параметра критичности, может привести к различным последствиям, начиная от усложнений условий полета и заканчивая гибелью людей.

3. Различные наборы функций в зависимости от режима функционирования (взлет/посадка, крейсерский полет, аварийный полет). Каждый из таких наборов требует оптимального распределения соответствующих ему функциональных задач с целью освобождения необходимых ресурсов или увеличения эффективности их использования.

4. Глубокая интеграция программной и аппаратной частей на всех этапах жизненного цикла системы. В соответствии с современными стандартами на проектирование ИМА поставщики аппаратных платформ и поставщики приложений на всех этапах проектирования должны гарантировать, что все ресурсы соответствуют требованиям целостности и готовности в соответствии с предоставленной им классификацией отказных состояний и данными по конфигурации системы.

Одной из последних работ в сфере оптимизации структуры многопроцессорных вычислительных систем автоматизированного управления в реальном времени является работа Терскова В.А. [64]. В данной работе был проведен анализ процессов функционирования систем данного типа и на его основе предложены следующие модели:

- модели оценки производительности МС;
- модели оценки надежности МС;
- модели функционирования МС.

Эти модели представляют собой графы состояний с параметрами вероятности нахождения системы в каждом из них.

На основе моделей была сформулирована задача выбора «эффективного варианта» МС по критериям производительности, надежности и стоимости. При этом

автором отмечается противоречивость критериев и необходимость использования метода прямого поиска (адаптивного поискового метода) для построения гибкой поисковой процедуры, допускающей настройку параметров оптимизации с использованием экспертных оценок для сокращения итераций перебора.

Предложенные аналитические и имитационные модели оказались очень неудобными и сложными с точки зрения создания алгоритмов оптимизации, поэтому автор переходит к алгоритмическому заданию критериев оптимального выбора и это ограничивает его в методах реализации (только метод прямого поиска).

Оптимизация структуры МС в работе [64] осуществлена с использованием модифицированного (имитационного) эволюционного алгоритма, суть которого заключается в наборе статистики результатов применения различных генетических операторов (скрещивания, мутации и др.) по отношению к рассматриваемым парам родителей. На основе этой статистики выбирается набор операторов, который приводит к созданию потомков с характеристиками, удовлетворяющими исходным требованиям.

Основным недостатком предложенного метода оптимизации применительно к ИМА является резко возрастающий объем статистики с ростом количества параметров оптимизации и сложность оценивания соответствия параметров поиска реальным требованиям к системе. Другими словами, с ростом сложности системы и с ростом количества параметров предложенный способ потребует большего количества вычислительных ресурсов при решении задачи оптимизации, а нахождение оптимального варианта не будет гарантировано. Кроме того, данный метод использует приведенные выше критерии оптимальности лишь как ограничения при оценке эффективности полученной популяции, в то время как в соответствии с современными стандартами на проектирование ИМА параметры надежности и производительности служат отправной точкой для выбора конечной архитектуры системы (при оценке функциональной опасности, предварительной оценке безопасности и назначении критичности функций).

Требования к повышенному уровню надежности также привели к появлению такого направления в проектировании систем как реконфигурация ресурсов. На

сегодняшний день проблема осуществления реконфигурации решается в двух направлениях:

- адаптация архитектуры вычислительной системы под структуру решаемой задачи;
- изменение параметров вычислительной системы при возникновении отказов.

Больших результатов в организации проблемно-ориентированных реконфигурирующихся многопроцессорных вычислительных систем в своих работах добились И.А. Каляев, И.И. Левин, Е.А. Семерников, В.И. Шмойлов и И.В. Беседин [35-37]. Их подход заключается в представлении потоковой задачи в виде множества векторов входных данных, которые обрабатываются по фиксированному алгоритму, изображенному в виде графа. При этом решением задачи является преобразование кортежа векторов входных данных в кортеж векторов выходных данных.

Повышение эффективности решаемой задачи предлагается осуществить путем организации так называемых «структурно-процедурных» вычислений, при которых исходный информационный граф разбивается на несколько независимых подграфов, каждый из которых реализуется на отдельных вычислительных ячейках, образующих «мультиконвейер». Структурность вычислений заключается в реализации структуры каждого подграфа на мультиконвейере, а процедурность – в организации определенной последовательности отображения этих подграфов в мультиконвейер.

Реконфигурация в этом случае осуществляется путем переключения ячеек мультиконвейера при изменении решаемой задачи на аппаратном уровне с помощью массивов ПЛИС, одновременно выполняющих функции вычислителей (см. рис. 1.16). То есть система в этом случае представляет собой объединение вычислительных блоков с помощью различных видов коммутации (ортогональной и иерархической).

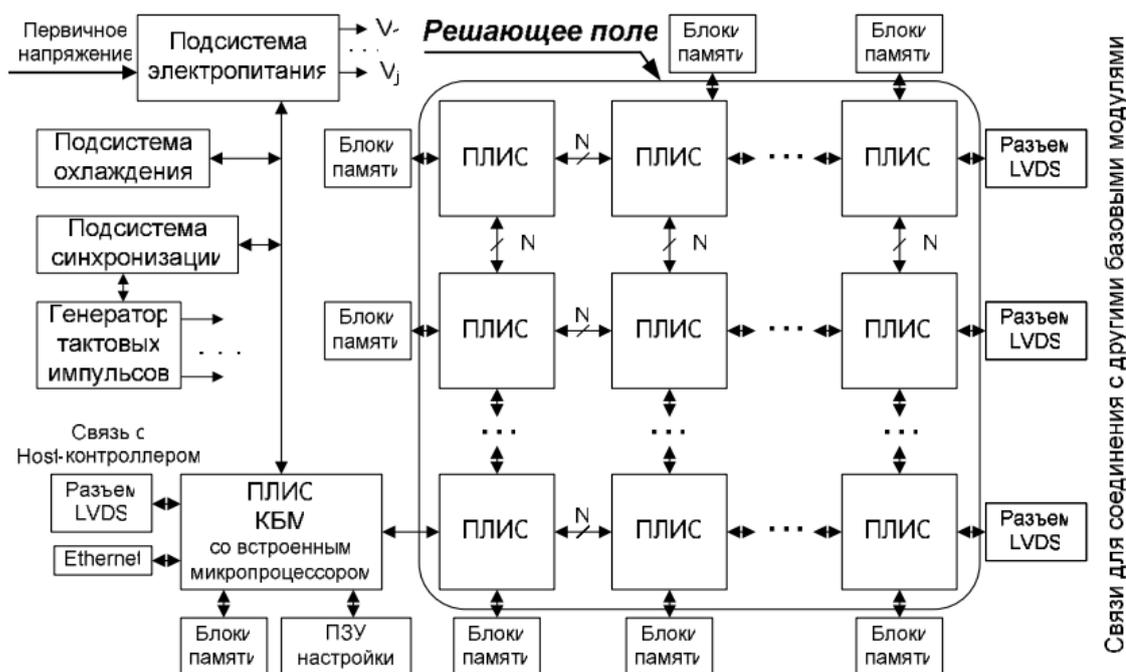


Рисунок 1.16 – Структура реконфигурируемой системы на основе ПЛИС [35]

С точки зрения структуры ИМА такая реализация реконфигурации потребует использования в качестве вычислителей ПЛИС, что приведет к необходимости реализации различных типов процессорных ядер для обеспечения возможности запуска на них копий ОСРВ. Кроме того, функциональные графы задач авионики будут настолько сложными, что выделить их в независимые подграфы вряд ли представляется возможным, а это, в свою очередь, лишает подобную архитектуру всех преимуществ при использовании в крейте ИМА.

Похожая реализация была предложена А.В. Лысенко и Г.В. Кремезом [65], но при этом ПЛИС не выполняют функции вычислителей.

Другой подход к реконфигурации МС на аппаратном уровне предложен В.В. Гролем, В.А. Романкевичем и С.М. Мораведжем в работе [66]. Он заключается в выделении из множества процессоров системы некоторого подмножества вычислителей, которые будут осуществлять частичное выполнение функций отказавшего элемента. Алгоритм распределения задач в случае отказа предусматривает использование таких параметров системы, как связность компонентов (топологии «кольцо», «последовательная цепочка», «каждый с каждым» и т.д.), количество отказавших и работоспособных процессоров, а также дополнительно время, затрачиваемое на решение задачи отказавшего элемента одним из «резервных».

Применительно к КБО ИМА такой подход обладает следующими недостатками:

- не принимаются во внимание параметры критичности функций отказавших и работоспособных элементов. При возникновении отказа процессор, выполняющий задачу с высоким уровнем критичности может быть дополнительно нагружен низкоприоритетной задачей, что приведет к ухудшению динамических характеристик системы за счет увеличения времени цикла обработки задачи с высоким приоритетом;

- не рассматривается возможность возникновения множественных отказов и, соответственно, необходимости исключения некоторых задач из состава КБО при нехватке резервных мощностей работоспособных элементов;

- не рассматривается реконфигурация при изменении решаемой задачи.

Наиболее актуальной в настоящее время и применимой к КБО ИМА реализацией принципов реконфигурации является «стратегия», разработанная А.А. Тарасовым.

Она основывается на выделении в US-системах (содержащих как специализированные S-модули, так и универсальные U-модули) трех видов деградации: *функциональной, структурной и структурно-функциональной* [67]. Функциональная деградация возникает при отказе какой-либо выполняемой функции без увеличения числа неисправных модулей. Структурная деградация предусматривает сохранение всех функций при отказах модулей. Структурно-функциональная деградация возникает при отказе одного или нескольких модулей с соответствующим уменьшением числа выполняемых функций.

В зависимости от видов деградации предлагаются так называемые «стратегии» функциональной перестройки системы. Данные стратегии подразумевают учет следующих параметров [68]:

- множества состояний системы, требующих перестройки;
- множества работоспособных и отказавших модулей;
- множества функций, связанных с работоспособными и отказавшими модулями;

- возможность исключения функций из состава системы или изменения алгоритмов их работы в сторону упрощения при возникновении отказа.

При этом рассматриваются все возможные комбинации сокращения числа выполняемых функций и упрощения алгоритмов их выполнения, а также возможные распределения этих комбинаций по модулям.

Формирование множества искомых вариантов вытекает в формальную постановку следующей задачи: найти множество вариантов распределения функций, обеспечивающих минимальные значения погрешности их выполнения для любого состояния системы при соблюдении ограничений на время их выполнения [67].

Применительно к КБО ИМА такие стратегии перестройки системы обладают следующими недостатками:

- функциональная деградация рассматривается лишь как *частичный* отказ какого-либо *модуля*, который приводит к снижению функциональных возможностей системы. То есть не учитываются отказы функций, которые могут возникать и при *полностью исправных модулях*;

- структурная деградация рассматривается как увеличение числа отказавших модулей, которое не оказывает влияния на функционирование системы [67]. Такой эффект может быть достигнут только при помощи резервирования или при построении аппаратной части на основе сильно избыточных модулей. Концепция же ИМА призвана увеличить показатели надежности при минимизации резервирования или полном его отсутствии;

- структурно-функциональная деградация для US-систем (ИМА также можно рассматривать как US-систему) рассматривается как отказ какого-либо *специализированного* модуля [67]. В ИМА такой отказ (при высоком уровне критичности соответствующей функции) будет означать потерю работоспособности всей системы. В представленных стратегиях указанная функция может передаваться на выполнение универсальному ВМ, что на практике для задач авионики практически невозможно из-за сложности специализированных задач (например, формирование изображения) и соответствующих аппаратных интерфейсов и архитектур вычислителей;

- стратегии перестройки при отказах направлены на такое распределение оставшихся функций, при котором обеспечивается их выполнение с наименьшей возможной погрешностью (за счет упрощенных алгоритмов вычисления) за минимально возможное суммарное время (в условиях увеличения времени обработки дополнительных назначенных на модуль задач). Из-за принадлежности ИМА к классу систем жесткого реального времени, суммарное время выполнения всех функций и некоторый резерв, которые образуют так называемый *интервал* времени (*gap*) T_{gap} , для комплекса будет постоянным. При исключении каких-либо функций из состава освободившееся время может быть передано другой функции с целью улучшения ее динамических характеристик. Следовательно, задача минимизации суммарного времени выполнения всех функций для систем ИМА не актуальна;

- при формировании стратегий перестройки задача оптимального распределения должна быть решена для всех возможных состояний аппаратной и функциональной частей (то есть необходимо осуществить полный перебор возможных вариантов конфигураций) [67]. Как следствие – экспоненциальный рост времени выполнения алгоритмов распределения с ростом числа функций и трудность масштабирования системы;

- стратегии перестройки не рассматривают различные режимы функционирования системы (с различными наборами функций).

1.4 Выводы

1. Обзор и анализ архитектуры, конструктивных особенностей и структурных элементов известных КБО ИМА показал, что, не смотря на существующие в настоящее время различные подходы к реализации основных положений проектирования интегрированных модульных комплексов, все они обладают общим набором характерных свойств:

- общей для комплекса аппаратной платформой;
- модульным составом;
- многозадачностью;
- изоляцией программных приложений друг от друга и от аппаратной платформы;
- централизованным размещением модулей;
- высокоскоростной сетью передачи данных;
- открытой архитектурой;
- возможностью замены модулей в условиях эксплуатации;
- возможностью подключения внешних интерфейсов к модулям.

2. В настоящее время существует множество методов проектирования аппаратной части интегрированных комплексов, в которых, однако, проблемы оптимизации распределения функциональных задач либо не рассматривают вовсе, либо рассматриваются с позиции независимой разработки программной части комплекса.

3. Структура перспективного отказоустойчивого КБО ИМА определяется тремя взаимосвязанными компонентами: аппаратной платформой, функциональными (программными) приложениями и ОСРВ – которые являются взаимодополняющими и должны проектироваться с учетом особенностей интеграции и совместной реализации.

4. Для построения аппаратной платформы комплексов ИМА широко используются COTS-компоненты, которые обеспечивают высокопроизводительную элементную базу при достаточной экономической эффективности. Для того чтобы

КБО ИМА, спроектированный с использованием подобных компонентов, не рассчитанных на работу при воздействии внешних факторов, соответствующих авиационной группе оборудования, удовлетворял требованиям авиационных норм и успешно прошел периодические и приемосдаточные испытания, необходимо при его разработке использовать специальные технологические и конструкторские методы.

5. Одним из наиболее перспективных конструктивных методов повышения надежности проектируемого КБО ИМА является совместное решение задач оптимального использования ресурсов вычислительных модулей, распределения функциональных задач и управления конфигурацией комплекса при возникновении отказов или изменении решаемых задач.

6. Для создания эффективных методов и алгоритмов проектирования КБО ИМА необходимо разработать специальную математическую модель комплекса, которая объединит в себе все его структурные составляющие и позволит получать, анализировать и оценивать различные архитектурные и структурные варианты построения комплекса.

2 РАЗРАБОТКА И ИССЛЕДОВАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ КОМПЛЕКСА БОРТОВОГО ОБОРУДОВАНИЯ ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИОНИКИ

Среди современных методов научного исследования [69-72] наиболее подходящим для изучения интегрированных КБО и авионики в целом является системный подход и неразрывно связанный с ним системный анализ, на основе которого можно осуществить синтез системы [73].

Применение системного подхода позволяет учесть следующие особенности бортового оборудования при проведении анализа (см. рис. 2.1):

- глубокая интеграция взаимодействующих между собой структурных элементов КБО ИМА и его вхождение в вышестоящие уровни иерархии в качестве подсистемы (взаимодействие с другими комплексами ЛА) представляет собой единого целого, состоящего из взаимодействующих между собой структурных элементов, так и в качестве подсистемы, входящей в вышестоящие уровни иерархии (комплексы ЛА);

- иерархическая структура комплекса, т.е. наличие различных уровней подчинения, на которых располагаются элементы системы (комплекс-крейт-модуль);

- ключевая роль связей между структурными элементами в функционировании комплекса ИМА и в предоставлении механизмов для перераспределения ресурсов;

- функциональная законченность структурных элементов комплекса и, как следствие, возможность использовать различные математические модели для каждого из них и для системы в целом.

Кроме того, обеспечение возможности динамического перераспределения ресурсов возможно только при детальном изучении всех видов взаимодействий элементов структуры проектируемого комплекса, а управление конфигурацией представляется как сложная задача, решение которой требует определения различных многомерных функций и критериев оптимальности. Ее решение возмож-

но только при системном рассмотрении КБО и создании математической модели не только отдельных его элементов, но и взаимодействий его структурных частей, а также алгоритмов реконфигурации.

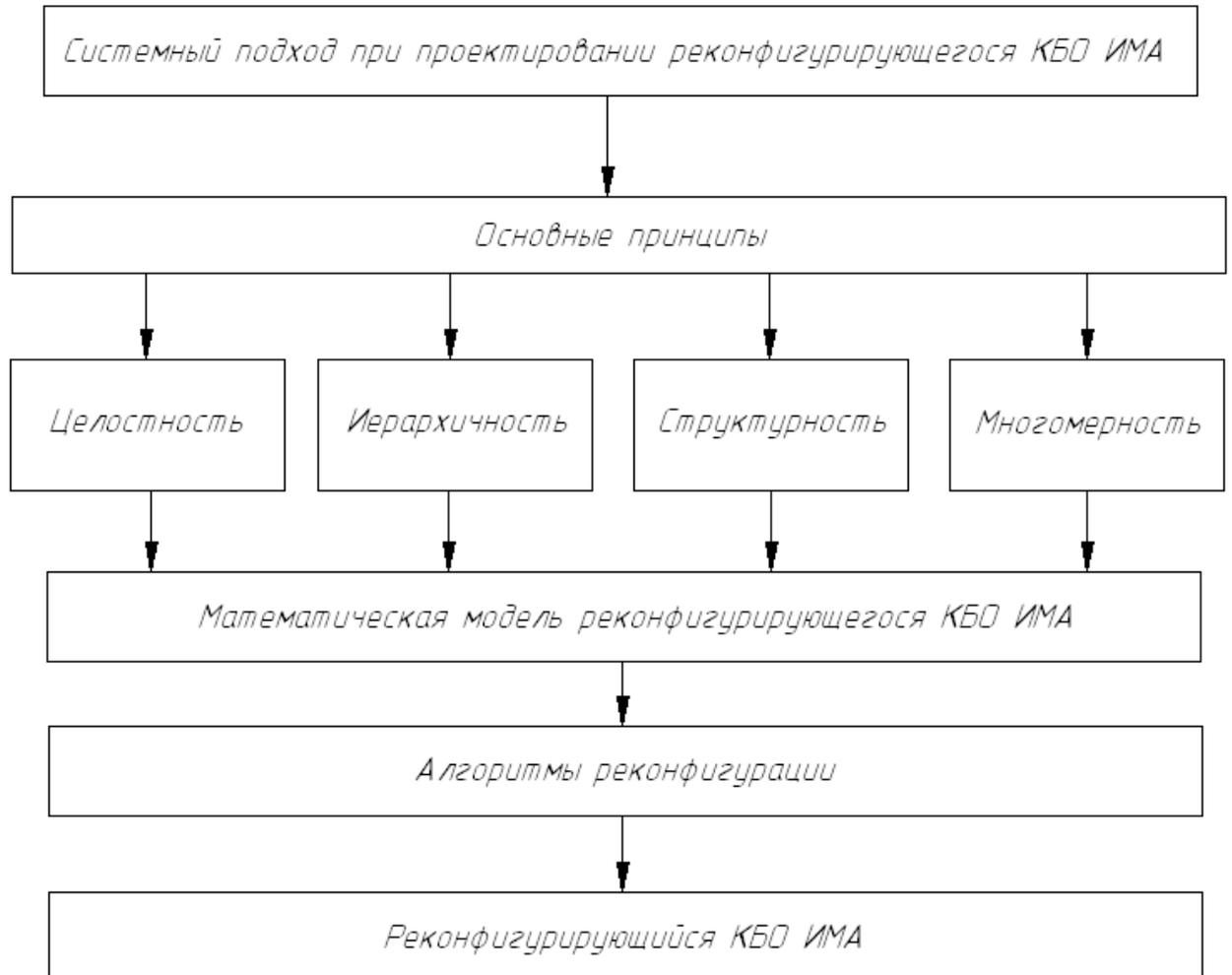


Рисунок 2.1 – Системный подход при проектировании реконфигурирующегося КБО ИМА

2.1 Математическое описание состава КБО ИМА

Математическое описание состава сложного программно-аппаратного КБО C представлено в виде трехмерного массива его конфигураций и основывается на трех составляющих, полученных в результате первичной декомпозиции: набор сменных КФМ S , на которых запущены программные приложения Y , работающие в различных режимах под управлением ОСРВ.

Каждое состояние f соответствует определенной конфигурации программных средств комплекса во время выполнения задачи, не затрагивающей его аппаратную или функциональную части и полученной, в общем случае, в результате *программной динамической реконфигурации*. При этом после проведения реконфигурации данного вида количество функционирующих приложений не меняется.

$$C_S = \{C_1(S_1, Y_1), C_2(S_1, Y_2), \dots, C_f(S_1, Y_m)\}, \quad (2.1)$$

где C_S – множество конфигураций КБО, которые могут быть получены в результате программной реконфигурации, S_l – количество исправных КФМ, Y_m – набор функционирующих программных приложений, соответствующий текущей конфигурации.

Каждое состояние n соответствует определенной конфигурации аппаратных средств комплекса во время выполнения задачи, не затрагивающей его функциональную часть и полученной, в общем случае, в результате *аппаратной динамической реконфигурации*. При этом после проведения реконфигурации данного вида количество функционирующих приложений меняется.

$$C_H = \{C_1(S_1, Y_1), C_2(S_2, Y_2), \dots, C_n(S_k, Y_r)\}, \quad (2.2)$$

где C_H – множество конфигураций КБО, которые могут быть получены в результате аппаратной реконфигурации, S_k – набор функционирующих КФМ, Y_r – набор программных приложений, оставленных в составе комплекса после проведения реконфигурации. При этом $k \in N, r \in N, k \neq r$.

Каждое состояние p соответствует определенной конфигурации аппаратных и программных средств при изменении выполняемой задачи, полученной, в общем случае, в результате *функциональной динамической реконфигурации*. При этом задействуются все исправные в данный момент КФМ, а набор функциональных приложений (их перечень и количество) могут изменяться.

$$C_T = \{C_1(S_k, Y_1), C_2(S_k, Y_2), \dots, C_p(S_k, Y_l)\}, \quad (2.3)$$

где C_T – множество конфигураций КБО которые могут быть получены в результате функциональной реконфигурации, Y_l – набор программных приложений для данного режима. При этом $l \neq m \neq r$, то есть функциональные наборы для каждого вида реконфигурации, в общем случае, будут различными.

Математическое описание КБО представляется в виде вектора $E(C)$, в котором координаты соответствуют состояниям комплекса после проведения различных видов реконфигурации:

$$E(C) = \{C_S, C_H, C_T\}. \quad (2.4)$$

В геометрическом выражении (см. рис. 2.2) математическое описание состава КБО ИМА представляется в виде поверхности, которая соответствует различным сценариям проведения всех видов реконфигурации. При этом ее форма зависит не только от параметров S_k , Y_m , Y_l , но и от выбранного критерия оптимальности (на рис. 2.2 не показано), по которому осуществляется распределение соответствующих ресурсов при выполнении той или иной реконфигурации.

Копии операционной системы в этой модели фактически представляют собой программные приложения, размещенные на аппаратных модулях и осуществляющие контроль над их исправной работой.

Выражения (2.1)-(2.4) определяют обобщенное описание состава реконфигурирующего КБО ИМА, которое будет использоваться при решении задач реконфигурации.

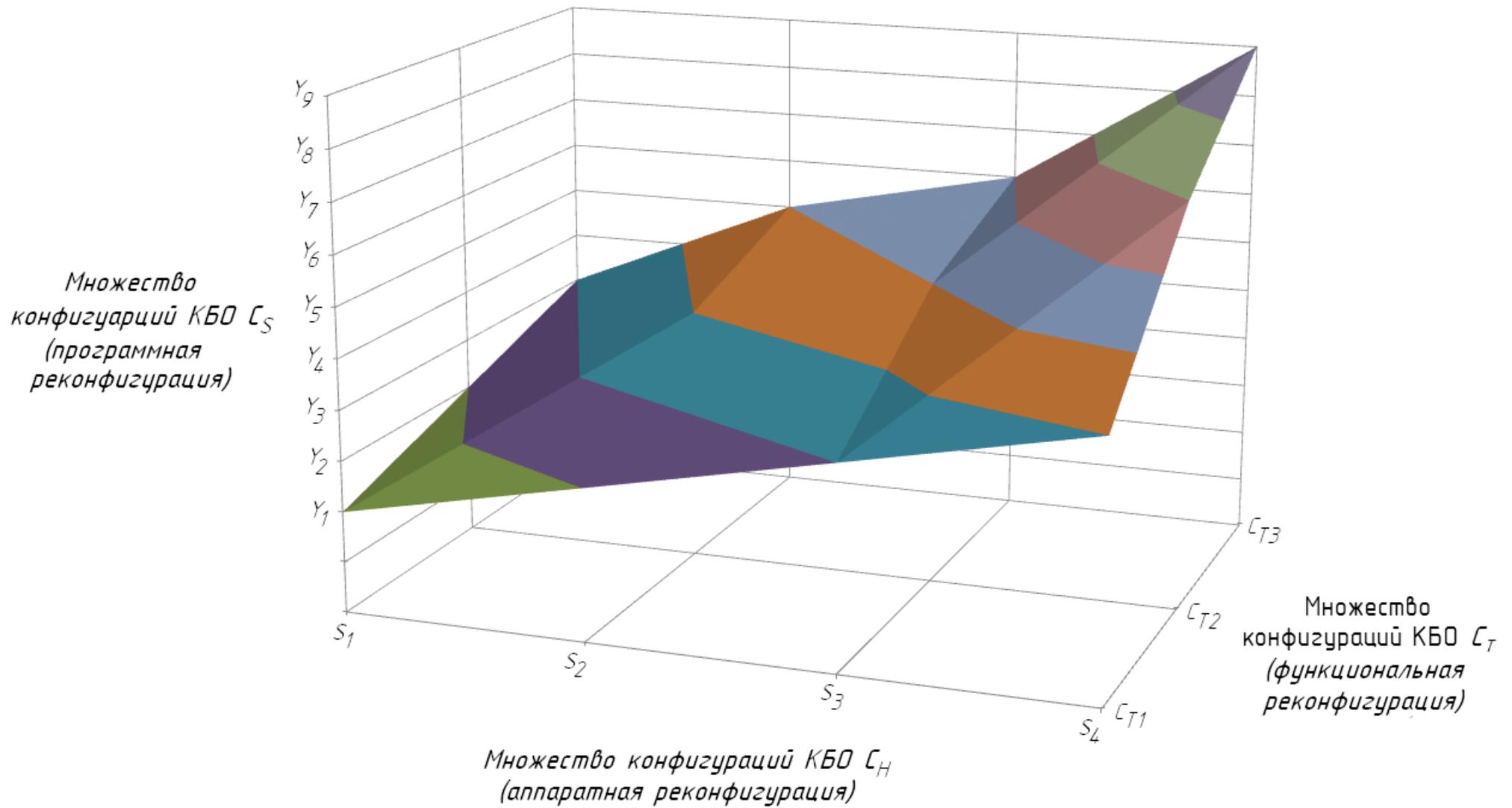


Рисунок 2.2 – Многомерная суперповерхность реконфигурирующегося КБО ИМА

2.2 Математическое описание структуры КБО ИМА

При проектировании комплексов авионики, даже при наличии утвержденного технического задания (ТЗ), целый ряд структурных характеристик разрабатываемой системы, как правило, определяется непосредственно разработчиком. К таким характеристикам относятся:

- модульный состав (как качественный, так и количественный);
- функциональный состав и распределение задач;
- организация каналов связи внутри комплекса.

Эти характеристики напрямую влияют на качество проектируемой системы, эффективность ее работы, удобство обслуживания, ремонтпригодность и стоимость. Решение об их реализации принимается на основе перечня функций, сформулированного в ТЗ. Каждой функции для реализации требуется определенный набор входных данных, получаемый в виде аналоговых сигналов или цифровых потоков данных от датчиков или смежных систем. Результат выполнения функции также передается в другие системы или функции проектируемого комплекса.

На основе анализа, проведенного в главе 1, аппаратную структуру КБО ИМА можно представить так, как показано на рисунке 2.3, то есть в виде набора КФМ, объединенных в сеть передачи данных при помощи коммутатора. Для описания функциональной структуры используется типовой набор функций, выполняемый КБО вертолета:

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;

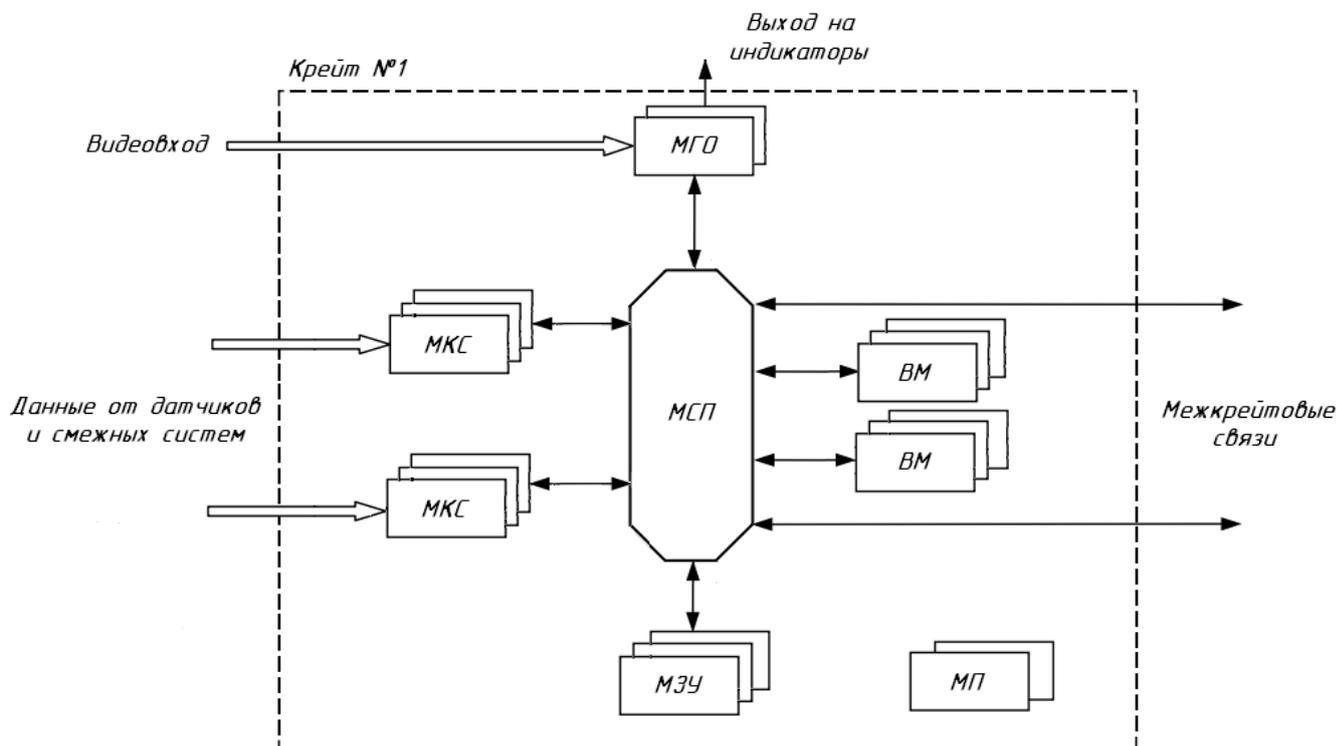


Рисунок 2.3 – Аппаратная структура КБО ИМА:

МКС – модуль-концентратор сигналов, МГО – модуль графической обработки, МСП – модуль сетевого переключателя (коммутатор), МЗУ – модуль запоминающего устройства, ВМ – вычислительный модуль, МП – модуль питания

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.

Пусть для реализации полного набора прикладных функций КБО необходимо выполнять множество Y приложений, работающих под управлением ОСРВ. В памяти модулей должно быть также размещено множество D массивов, представляющих собой исходные данные и результаты операций обработки. Заданы размеры σ_d массивов $d \in D$ и приложений σ_y , $y \in Y$. В σ_d и σ_y учтен также размер рабочей памяти, необходимый для выполнения приложениями операций обработки информации. Известны частота ν_y обращения приложения y к тому или иному массиву данных и объем вычислений ρ_y , необходимый для реализации приложения (например, в FLOPS). Возможным вариантом представления связей между функциями рассматриваемого КБО может выступать таблица 2.1. Порядковые номера приложений y соответствуют порядковым номерам из набора функций.

Функциональную структуру комплекса можно представить в виде ориентированного графа $G_F = (Y, Z)$, в котором каждому приложению или массиву ставится в соответствие вершина $y \in Y$, а информационной связи между ними – дуга $z \in Z$. В этом графе изображение вершин, соответствующих массивам данных, не производится, а заменяется дополнительными дугами z , инцидентными соответствующим вершинам y , поскольку они неразрывно связаны с функциями-источниками и функциями-приемниками. Пример функционального графа G_F рассматриваемого КБО представлен на рисунке 2.4.

Пусть $k \in K$ – параметр критичности функции $y \in Y$, которая входит в состав ориентированного графа функциональной структуры G_F . В соответствии с [60], 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. Процесс назначения уровня критичности какому-либо элементу комплекса ИМА является трудоемким и итеративным процессом с доказательством правильности присвоенного уровня в соответствии с Р-297 [58] и КТ-254 [60].

Таблица 2.1 – Функции КБО вертолета и массивы данных

$y \in Y$	Программные приложения $y \in Y$																
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}	y_{17}
y_1		■	■	■	■	■		■	■					■			
y_2			■		■			■						■			
y_3																	
y_4																	■
y_5			■														
y_6	■		■				■							■			
y_7	■		■														
y_8			■														
y_9																	
y_{10}																	
y_{11}			■														■
y_{12}																	■
y_{13}											■	■	■	■			■
y_{14}																	
y_{15}			■							■							
y_{16}															■		
y_{17}			■													■	

Примечание – закрашенная область соответствует наличию информационной связи между функциями

Таким образом, математическое описание функциональной структуры КБО ИМА представляет собой ориентированный граф, в узлах которого находятся функциональные приложения, характеризующиеся параметрами критичности и объема вычислений, которые соединены дугами, соответствующими массивам данных с параметрами частоты обращения и размера:

$$G_F = (Y(K, P), Z(\Xi, \vartheta)), \quad (2.5)$$

где P – необходимый объем вычислений для функции, Ξ – объем массива данных, ϑ – частота обращения к массиву.

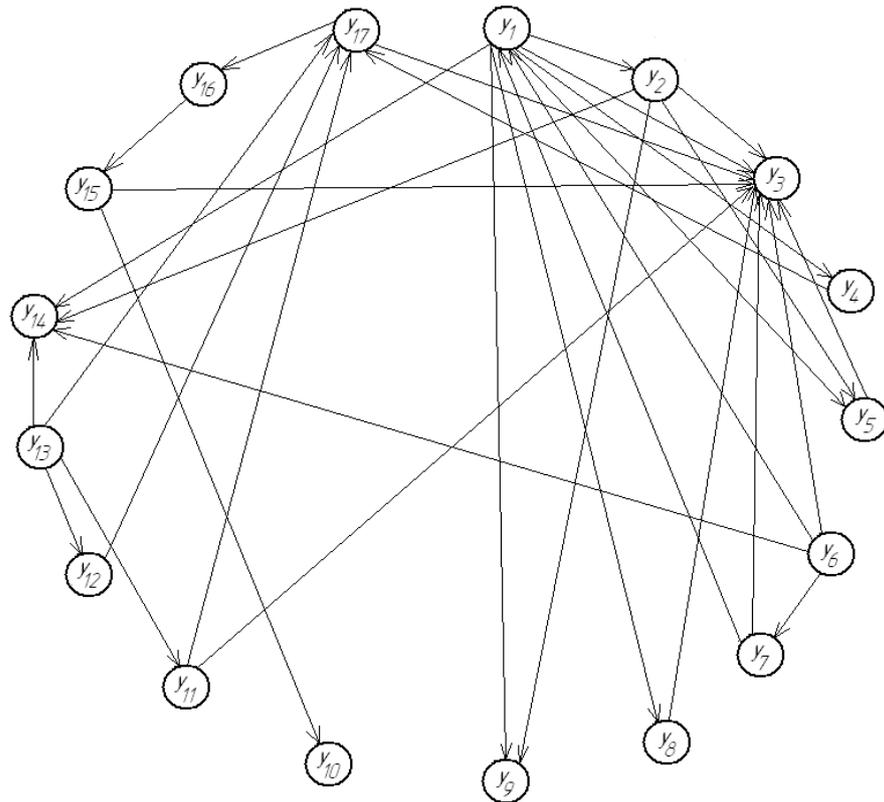


Рисунок 2.4 – Функциональный граф КБО вертолета

Поскольку аппаратная платформа в виде модульного состава и каналов передачи данных в интегрированном комплексе неразрывно связана с программными приложениями (хоть и изолирована от них посредством ОСРВ), то математическую модель аппаратной структуры целесообразно строить на тех же принципах, что и функциональной.

Так как весь набор функций $Y(K,P)$, независимо от их специфики, функционирует на модулях, входящих в крейт, а массивы данных передаются по межмодульным каналам связи, то аппаратную структуру удобнее всего также представить в виде ориентированного графа $G_S=(V, W)$, где вершины $v \in V$ соответствуют модулям, а ребра $w \in W$ – каналам связи между ними.

Каждый модуль обладает рядом параметров:

- степень загрузки λ_v , памяти модуля $v \in V$, которая определяется размерами массивов данных, приложений и рабочей памяти, необходимой для выполнения операций приложений, размещенных в этом модуле;

- емкость памяти модуля L_v , необходимая для выполнения приложений и хранения массивов данных;

- производительность процессора R_v .

Обозначим через $\psi_{n,m}$ передаваемое в среднем количество данных в единицу времени между парой модулей $n, m \in V$. Тогда пропускная способность магистрали ω для сети с парными связями должна будет удовлетворять следующему условию [74]:

$$\alpha_\omega \beta_\omega \psi_{n,m} \leq \Omega, \forall (n,m) \in V, n \neq m, \quad (2.6)$$

где α_ω – коэффициент запаса, учитывающий метод доступа, применяемый в данной сети (например, снижение реальной пропускной способности из-за особенностей протокола обмена), β_ω – коэффициент запаса, учитывающий неравномерность загрузки канала связи и определяемый на основе экспертных оценок.

Таким образом, математическое описание аппаратной структуры КБО ИМА представляет собой ориентированный граф, в узлах которого находятся модули (независимо от их типа), характеризующиеся параметрами емкости памяти и производительностью процессора, которые соединены дугами, соответствующими каналам передачи данных с параметрами пропускной способности и объема передаваемых данных в единицу времени:

$$G_S = (V(L, R), W(\Psi, \Omega)). \quad (2.7)$$

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». Это позволяет не учитывать параметр $\psi_{n,m}$ при решении задачи распределения функций.

2.3 Математическая модель КБО ИМА.

Алгоритм распределения функциональных задач

В общем случае функционирование вычислительной системы ИМА характеризуется тремя показателями загрузки: сети передачи данных, процессоров и памяти. Каждый из этих показателей зависит от размещения приложений и массивов данных в системе. Поэтому КБО ИМА может быть оптимизирован либо по одному из показателей загрузки (два других в этом случае при решении задачи вводятся в ограничения), либо проводится многокритериальная оптимизация по всем трем показателям. В данной работе рассматривается частный алгоритм размещения приложений по минимуму критерия загрузки сети передачи данных.

Размещение приложений и массивов данных по минимуму загрузки сети позволяет уменьшить время, затрачиваемое на обмен данными между модулями по каналам связи, а, следовательно, время реализации и запаздывания прикладных функций (что является критичным фактором в некоторых приложениях). При этом значительно повышается надежность выполнения функций ввиду уменьшения времени использования и числа обращений к каналам связи. При решении задачи по указанному критерию и при наличии каналов связи заданной пропускной способности снижение загрузки каналов позволяет повысить достоверность передачи данных путем введения избыточности, а также увеличивает надежность выполнения функций. С точки зрения динамического управления конфигурацией комплекса размещение приложений с наибольшим значением суммарного потока информации между ними обеспечит полное (или частичное) выполнение критичных функций в случае отказа сетевого коммутатора.

В [22] предлагается методика декомпозиции вычислительной системы на функциональные элементы, которая заключается в выделении независимых подграфов из многосвязного графа операций обмена и обработки информации. При этом предполагается реализация этих подграфов на отдельных вычислителях. Однако такой подход не учитывает специфику организации вычислительно процесса в ОСРВ. Поскольку каждая прикладная функция реализована в виде сложного

программного приложения, которое не может выполняться одновременно на двух вычислителях, то нет необходимости в выделении независимых частей из множества функций и операторов (такими частями являются сами приложения). Необходимо также принимать во внимание и структуру крейта ИМА, в котором данные могут передаваться как от одного ВМ к другому, так и посредством общей памяти МЗУ. Это зависит, прежде всего, от выбранного типа многопроцессорной системы. Кроме того, в сложных системах с множеством функциональных связей выделение независимых подмножеств будет крайне проблематично.

Поскольку КБО ИМА работает под управлением ОСРВ, а прикладные функции (самолетовождение, управление общесамолетным оборудованием и др.) реализованы в виде отдельных приложений разного уровня сложности и критичности, разделяющих общие вычислительные и информационные ресурсы [38], то актуальной является задача разработки математических методов оптимального распределения этих приложений между доступными аппаратными ресурсами. Задача размещения актуальна и в ходе функционирования системы, например при вводе новых прикладных функций и соответствующих им программных приложений. Необходимость в перераспределении возникает также при выходе из строя отдельных модулей в крейте. В этих случаях правильное решение проблемы размещения будет способствовать поддержанию работоспособности комплекса и выполнению им поставленной задачи, а также обеспечит возможность динамического управления конфигурацией программных и аппаратных ресурсов. Это, в свою очередь, минимизирует резервирование и гарантирует максимально возможный уровень безопасности и эффективности завершения полета или его дальнейшего выполнения в случае отказов [38].

Решение задачи размещения основывается на математических моделях функциональной и аппаратной структур (2.5) и (2.7). Дополнительно определены значения следующих булевых переменных:

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

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

Загрузка λ_v памяти модуля $v \in V$ определяется размерами массивов данных, приложений и рабочей памяти, необходимой для выполнения операций приложений, размещенных в этом модуле:

$$\lambda_v = \sum_{d \in D} \sigma_d x_{dv} + \sum_{y \in Y} \sigma_y x_{yv}. \quad (2.8)$$

При решении задачи должны быть соблюдены следующие ограничения:

- емкость памяти модуля L должна быть достаточной для выполнения приложений и хранения массивов данных, размещенных в нем:

$$\sum_{d \in D} \sigma_d x_{dv} + \sum_{y \in Y} \sigma_y x_{yv} \leq L_v, \forall v \in V, \quad (2.9)$$

- быстродействие процессора R каждого модуля должно быть достаточным для выполнения с соответствующей частотой всех приложений, размещенных в модуле:

$$\beta_v \sum_{y \in Y} v_y \rho_y x_{yv} \leq R_v, \forall v \in V, \quad (2.10)$$

где β_v – коэффициент запаса на основе экспертных оценок.

Кроме того, каждое приложение должно выполняться на одном ВМ [71]:

$$\sum_{y \in Y} x_{yv} = 1, \forall y \in Y. \quad (2.11)$$

Каждый массив должен храниться в памяти хотя бы одного модуля:

$$\sum_{v \in V} x_{dv} \geq 1, \forall d \in D. \quad (2.12)$$

Выражения (2.9) – (2.12) определяют условия решения задачи размещения без учета архитектуры модулей крейта.

Пусть функциональная структура КБО описывается в виде ориентированного графа $G_F=(Y, Z)$, а аппаратная структура - в виде ориентированного графа $G_S=(V, W)$. Тогда проблему размещения приложений и массивов данных в модулях можно рассматривать как поиск отображения множества вершин графа G_F на множество вершин графа G_S [75, 76].

Пример функционального графа G_F КБО представлен на рисунке 2.4. Требуемая производительность R_v процессора (которая может быть получена на основе анализа количества подфункций и их сложности) для выполнения соответствующей функции определяется в соответствии с их порядковыми номерами, т.е. от R_{v1} до R_{v17} . В дальнейшем будем полагать, что процессор каждого ВМ обладает быстродействием R_w . Помимо ВМ в крейте размещены специализированные модули (МКС, МЗУ, МГО), и их аппаратная реализация отличается от базовых ВМ, поэтому определить соответствующие им приложения в какой-либо другой модуль без изменения аппаратной структуры крейта невозможно. Значит, при распределении некоторых прикладных функций необходимо учитывать невозможность их совмещения с другими функциями. Однако в аварийной ситуации технически возможно назначить некоторые функции базового ВМ в модули МКС и МЗУ, если соблюдаются ограничения (2.9) и (2.10), но такой режим работы не рассматривается.

Для отображения множества вершин графа G_F на множество вершин графа G_S по какому-либо критерию оптимальности могут использоваться алгоритмы решения типовых задач, связанных с оптимизацией сетей и графов, которые отличаются в основном набором накладываемых ограничений и временем реализации. При этом выбор задачи зависит от целей, которые ставит перед собой разработчик КБО.

Задача поиска максимального потока, применительно к распределению ресурсов комплекса, состоит в поиске способа пересылки максимального количества единиц потока информации из модуля/приложения источника в модуль/приложение приемник при условии отсутствия превышения показателей пропускной способности сети передачи данных.

Для решения этой задачи, как правило, используются следующие исходные данные и ограничения [77]:

- определение вершины-источника y_s и вершины-приемника y_t ;
- для любого потока из y_s в y_t количество данных, выходящих из любой вершины $y \in Y$ (кроме $y = y_s$ и $y = y_t$) должно быть равно количеству данных, входящих в нее;
- количество данных, проходящих по каждой дуге, не должно превышать пропускной способности этой дуги;
- суммарный объем данных, выходящих из вершины-источника y_s , должен быть равен суммарному объему данных, входящих в вершину-приемник y_t .

Существует несколько алгоритмов решения этой задачи. Метод линейного программирования (симплекс-метод) в данном случае является слишком громоздким, и его сложность растет по экспоненциальному закону. Более удобными представляются алгоритмы Форда-Фалкерсона, Эдмондса-Карпа и Диница, которые заключаются в поиске увеличивающейся цепи и увеличении потока вдоль найденной цепи до максимального значения с последующим поиском новой увеличивающейся цепи. Если на каком-то этапе поиска увеличивающую поток цепь найти не удастся, то алгоритм заканчивается, а текущий поток из y_s в y_t является максимальным. Различия между этими алгоритмами заключаются лишь в модификации процедуры поиска увеличивающейся цепи.

Применение этих алгоритмов в «чистом» виде для рассматриваемой модели КБО сильно ограничено по следующим причинам:

1. Алгоритмы работают только для целых значений пропускных способностей. Иначе они могут выполняться бесконечно долгое время и не сходятся к верному решению.

2. Необходимость определения вершины-источника и вершины-приемника, что для рассматриваемой модели не имеет смысла. Даже если в графах структур КБО и окажутся вершины, в которые не входит (или из которых не выходит) ни одна дуга, то это не означает, что они служат приемниками (источниками) всей информации, проходящей по графу. Более того, таких источников (приемников)

может быть несколько. Искусственное добавление единых стока и истока, а также ребер с бесконечной пропускной способностью, позволит лишь найти максимальный поток, но не распределить соответствующие ему функции.

3. Структура КБО описывается двумя графами. Применение приведенных алгоритмов к графу G_F с последующим определением вершин с максимальным потоком данных на вершины графа G_S теоретически возможно, но они не учитывают ограничения (2.9) – (2.10).

Таким образом, задача поиска максимального потока (а также ее разновидность – задача поиска потока минимальной стоимости) данных может быть лишь частично применена к проблеме оптимизации передачи данных в структуре КБО ИМА, представленной в виде графов. Необходима как модификация постановки задачи и исходных данных, так и соответствующих алгоритмов решения.

Частным случаем транспортной задачи (как и поиск максимального потока) является *задача о назначениях* [78]. Она заслуживает отдельного рассмотрения из-за схожести своей структуры со структурой задачи распределения приложений по модулям в КБО.

Исходными данными для данной задачи служат два множества Y и V , функция стоимости $C: Y \times V \rightarrow R$. При этом необходимо найти биекцию $f: Y \rightarrow V$, такую, что минимальная следующая целевая функция

$$\sum_{y \in Y} C(y, f(y)). \quad (2.13)$$

Наиболее известным из множества алгоритмов решения этой задачи является венгерский алгоритм с временной сложностью $O(n^4)$ (существует модифицированный алгоритм со временем выполнения $O(n^3)$). Он основывается на применении двудольного графа, к которому можно свести полученные графы структур комплекса G_F и G_S .

Как и задача о поиске максимального потока задача о назначениях и алгоритмы ее решения обладают существенными недостатками при применении к рассматриваемой модели комплекса:

1. Множества Y и V должны быть одного размера, что для комплекса авионики невозможно из-за особенностей архитектуры и физических ограничений. Алгоритмы могут быть модифицированы под множества разного размера, но это потребует добавления (в нашем случае) «фиктивных» модулей для выполнения множества задач, что приведет к большой размерности матрицы назначения.

2. Как следствие, большое время решения задачи при росте количества выполняемых комплексом функций. Например, при 30 функциях и 30 аппаратных модулях (25 из которых «фиктивные») необходимо будет перебрать $30^4 = 810\,000$ возможных вариантов при применении оригинального венгерского алгоритма.

3. Минимум функции $\sum_{y \in Y} C(y, f(y))$ отражает лишь одно из ограничений, описанных в выражениях (2.9) – (2.10). Модификация соответствующих оригинальных алгоритмов под множество ограничений увеличит их сложность и время выполнения.

Распределение приложений по модулям может также рассматриваться как типовая одномерная задача об упаковке, которую можно применить по отношению к графам функциональной и аппаратной структур. Необходимо разместить функции по ВМ так, чтобы как можно более равномерно распределить нагрузку на процессоры. Для решения этой задачи, применительно к рассматриваемому КБО, используются следующие исходные данные:

- множество модулей V с производительность R_v процессора;
- множество функций Y с параметрами необходимой для работы производительностью процессора ρ_y .

Необходимо найти целое число модулей B и разбиение множества Y на B подмножеств $S_1 \cup \dots \cup S_B$, таких что

$$\sum_{i \in S_B} y_i \leq V, k = 1, \dots, B.$$

При этом решение называется оптимальным, если B минимально.

Данная задача относится к классу NP-полных задач [79], а все алгоритмы ее решения имеют экспоненциальную временную сложность [80, 81], что равно-

сильно полному перебору всех возможных вариантов. Применение известных алгоритмов (Best fit decreasing – лучший, подходящий по убыванию, и First fit decreasing – первый, подходящий по убыванию) в «чистом» виде для рассматриваемой модели и КБО в целом невозможно по следующим причинам:

1. Даже при малом количестве функциональных задач (порядка нескольких десятков) время работы алгоритмов решения задачи будет измеряться годами, даже при использовании мощнейших компьютеров [82].

2. Как следствие, невозможность масштабирования системы при исключении существующих или включении новых задач из-за необходимости пересчета по алгоритму.

3. Необходимость учета дополнительных ограничений и трудность адаптации к ним существующих алгоритмов решения.

Следовательно, для распределения функциональных задач по аппаратной платформе КБО нужно создать алгоритм, который за короткое время находит приемлемое решение и адаптируется под дополнительные ограничения. При этом можно частично использовать входные данные и методы решения типовых задач, приведенных выше.

Для постановки задачи распределения требуется описать общую математическую модель КБО ИМА, которая объединяет его аппаратную и функциональную структуры, и соответствующие им математические описания.

Модель представляется в виде двудольного графа, состоящего из графов G_F и G_S (см. рисунок 2.5). Его отличие от графа задачи об упаковке заключается в наличии связей не только между долями, но и внутри самих долей, которые впоследствии используются при формировании целевой функции и ограничений. Связи между графами G_F и G_S определяются выбранным критерием оптимальности. *Целевая функция* обеспечения минимума загрузки сети представлена в выражении (2.14) и формулируется следующим образом: распределение функций Y модулям V должно быть таким, чтобы обеспечить минимальный поток данных между модулями с учетом ограничений (2.9) и (2.10):

$$\min \left(\sum_{v \in V} \sigma_v \nu_v \right), \quad (2.14)$$

где $\sigma_v \nu_v = \sum_{m \in M} \sigma_m \nu_m - \sum_n \sum_{y \in Y_n} \sigma_y \nu_y$, при чем $\sum_{v \in V} \sigma_v \nu_v$ – поток данных между модулями после распределения задач, $\sum_{m \in M} \sigma_m \nu_m$ – суммарный поток данных между функциями до распределения, $\sum_n \sum_{y \in Y_n} \sigma_y \nu_y$ – суммарный поток данных между функциями после распределения, Y_n – набор функций, закрепленный за модулем n .

С целью сокращения затрат на разработку аппаратной платформы и исключения необходимости доказательства соответствия всех ее элементов самому высокому уровню гарантии конструирования аппаратуры по КТ-254 в модель необходимо дополнительно ввести параметр критичности ВМ, который принадлежит тому же множеству, что и параметр критичности функций ($k \in K$). В результате к ограничениям (2.9) и (2.10) добавляется дополнительная проверка на соответствие уровней критичности ВМ и приложений при их назначении.

Критерий оптимальности и ограничения для данной модели могут быть выбраны в соответствии с требованиями к системе. При этом задача распределения определяется как минимизация (максимизация) целевой функции с учетом ограничений. Для ее решения предлагается следующий алгоритм, представляющий собой глубоко модифицированный алгоритм решения задачи об упаковке, с целью возможности накладывания дополнительных ограничений и отсутствия необходимости осуществления полного перебора всех возможных комбинаций.

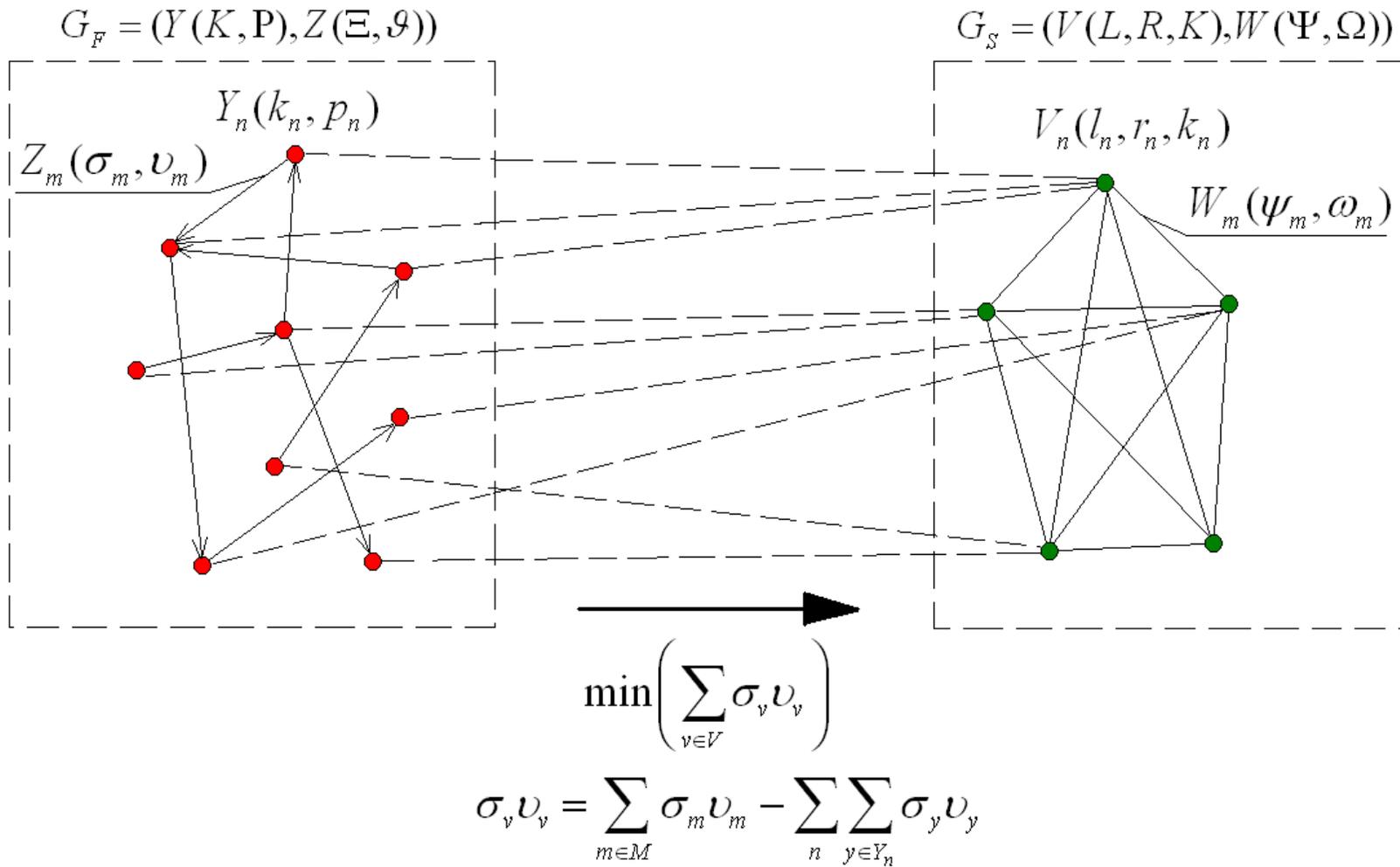


Рисунок 2.5 – Математическая модель КБО ИМА в виде двудольного графа

На *первом этапе* выбираются функции, требующие реализации на специальном вычислителе или вне крейта ИМА [83], в рассматриваемом случае это:

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

На *втором этапе* выбираются функции, требующие для своей работы всю вычислительную мощность процессора ВМ, то есть функции, удовлетворяющие условию:

$$v_y \rho_y = R_w, y \in Y.$$

Для КБО, представленного на рисунке 2.4, это следующие функции:

- 3. Формирование и отображение ПНИ;
- 4. Пилотирование и управление полетом;
- 5. Круглосуточная обзорная система.

Приведенные выше функции (вершины), а также инцидентные им информационные дуги, на время исключаются из функционального графа и не участвуют в дальнейшем распределении. С учетом этого граф принимает вид, представленный на рисунке 2.6. Для обозначения дуг используются значения потоков данных $v_m \sigma_m$, поскольку фактически по данному каналу связи передается массив данных размером σ_m с частотой обращения к нему v_m , где $m \in M$ – количество потоков данных.

Для удобства дальнейших рассуждений потоки данных упорядочены по возрастанию следующим образом [83]:

$$\begin{aligned} &v_{1,2}\sigma_{1,2} > v_{13,12}\sigma_{13,12} > v_{1,8}\sigma_{1,8} > v_{1,9}\sigma_{1,9} > v_{1,14}\sigma_{1,14} > v_{13,11}\sigma_{13,11} > \\ &> v_{2,14}\sigma_{2,14} > v_{15,10}\sigma_{15,10} > v_{2,9}\sigma_{2,9} > v_{12,17}\sigma_{12,17} > v_{11,17}\sigma_{11,17} \\ &> v_{13,14}\sigma_{13,14} > v_{13,17}\sigma_{13,17} > v_{7,1}\sigma_{7,1} > v_{7,17}\sigma_{7,17}. \end{aligned} \quad (2.15)$$

Первая цифра указывает на функцию, которая передает массив данных, а вторая – на функцию, которая принимает этот массив.

Также полагается, что объем памяти $l_n \in L$ каждого ВМ достаточен для хранения любого набора функциональных задач, т. е. условие (2.9) выполняется всегда.

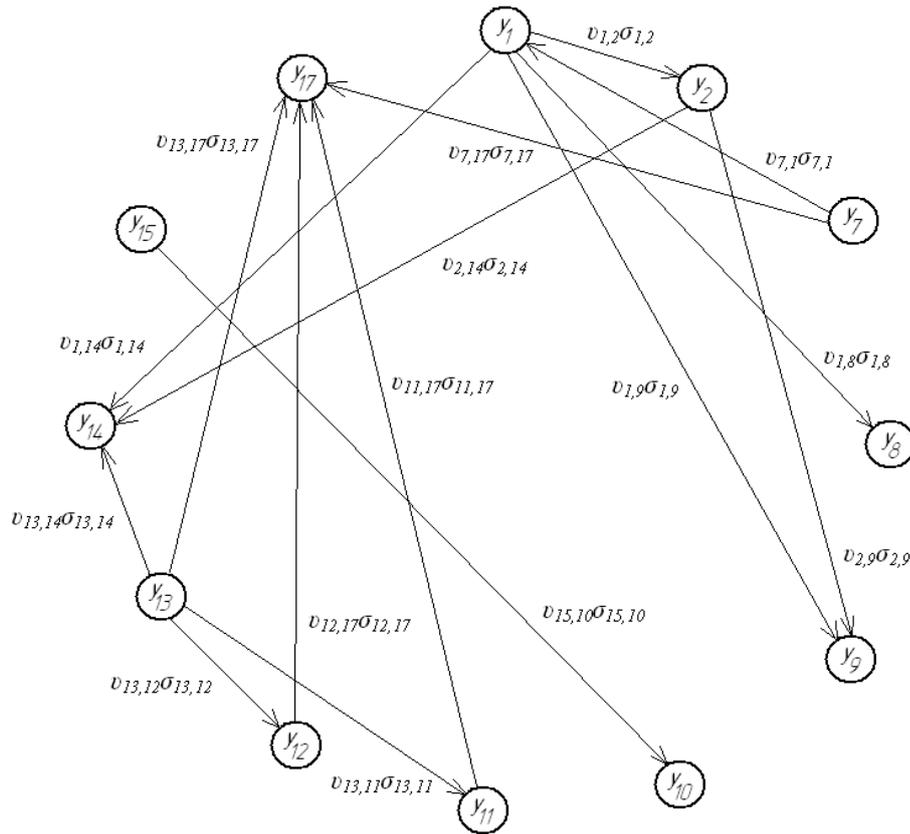


Рисунок 2.6 – Функциональный граф подсистемы КБО ИМА [83]

Алгоритм поиска размещения по критерию минимума загрузки сети передачи данных (2.14) заключается в последовательном размещении функций по модулям при условии соблюдения ограничений по емкости памяти, быстродействию процессоров и соответствию уровней критичности ВМ и приложений. Выбор данного критерия в отношении КБО ИМА обусловлен тем, что он представляет собой систему жесткого реального времени, где задержки в передаче данных могут привести к гораздо более тяжелым последствиям нежели, например, полная загрузка процессора или памяти при проведении реконфигурации. А также из соображений надежности передачи данных к приложениям с высоким уровнем критичности для сохранения их полного (или частичного) функционирования в случае отказа сети.

Алгоритм состоит из трех шагов [83]:

Шаг 1. Из множества всех вершин $y \in Y$ выбирается пара с максимальным значением потока данных $v_m\sigma_m$.

Шаг 2. Проверяется возможность реализации выбранной пары приложений на отдельном ВМ, т. е. не нарушаются ли ограничения (2.9) и (2.10). Если ограничения не нарушаются, то выбранные вершины отображаются на свободную вершину графа G_S с параметром k , соответствующим самой критичной функции из выбранной пары. Если же одна из вершин, принадлежащих выбранной паре, уже закреплена за какой-либо вершиной графа G_S , то проверяется возможность отображения второй вершины пары на эту же вершину графа G_S с учетом параметра k . Если и в том и в другом случае либо не выполняются условия (2.9) и (2.10), либо параметры k не соответствуют друг другу, то выбранную пару оставляют неотображенной.

Шаги 1–2 выполняются циклически (при этом каждый раз выбираются две вершины со следующим по порядку меньшим значением потока данных) до тех пор, пока все вершины графа G_F не будут отображены на вершины графа G_S или же пока не останется одна не отображенная вершина, либо несколько неотображенных вершин, которые нельзя реализовать на одном вычислителе (из-за особенностей построения системы или из-за несоблюдения ограничений (2.9) и (2.10)).

Шаг 3. Оставшиеся вершины графа G_F отображаются на свободные вершины графа G_S в соответствии с их параметрами критичности k . Конец вычислений.

Таким образом, алгоритм состоит из последовательного перебора пар вершин графа G_F по убыванию значения потока данных между ними и отображения указанных пар на вершины графа G_S при соблюдении ограничений на объем памяти и быстродействие процессора.

При исключении из алгоритма процедуры сопоставления параметров k ВМ и приложений количество модулей с высоким требуемым уровнем гарантии конструирования возрастет. Однако стоит отметить, что присвоение максимального значения k для всех модулей является более предпочтительным (хотя и более дорогим) с точки зрения управления конфигурацией, так как в соответствии со стандартами Р-297, КТ-178В и КТ-254 нельзя назначить критичную функцию на ВМ с более низким уровнем гарантии конструирования. Менее критичные функ-

ции, как правило, более просты в реализации и менее требовательны к ресурсам аппаратной части, что позволяет проектировать соответствующие им ВМ с низким уровнем гарантии аппаратуры, закладывая в них меньшую производительность. При таком подходе ограничения (2.9) и (2.10) расщепляются на несколько идентичных ограничений, каждое из которых соответствует выбранному уровню k ВМ и должно проверяться при реализации алгоритма. Это также ограничивает возможности реконфигурации в перспективе.

Для выполнения алгоритма в вычислителе необходимы следующие исходные данные:

- матрица смежности $A(G_F)=[a_{ij}]$, построенная на основе графа функциональной структуры и таблицы 2.1, где

$$a_{ij} = \begin{cases} 1, (y_i, y_j) \in Z \\ 0, (y_i, y_j) \notin Z \end{cases}$$

- матрица потоков данных $D(G_F)=[c_{ij}]$, где

$$c_{ij} = \begin{cases} v_{ij}\sigma_{ij}, (y_i, y_j) \in Z \\ 0, (y_i, y_j) \notin Z \end{cases}.$$

Можно использовать две аналогичные матрицы с разделенными параметрами v_m и σ_m ;

- набор ограничений, которые должны выполняться в процессе распределения, записанные в виде системы неравенств, например

$$Cond = \begin{cases} \sum_{y \in Y} \sigma_y x_{yv} \leq L_v, \forall v \in V \\ \dots \\ \sum_{y \in Y} \rho_y x_{yv} \leq R_w, \forall v \in V \end{cases}.$$

- в качестве исходных данных может задаваться количество доступных модулей. При этом возможны варианты, при которых распределение по указанному критерию либо не будет осуществлено (недостаточное количество модулей), либо оно завершится со свободными модулями в остатке.

Выходными данными алгоритма служит прямоугольная матрица $R(G_s)=[b_{ij}]$, построенная на основе графа аппаратной структуры, где

$$b_{ij} = \begin{cases} 1, & y_j \in V_i \\ 0, & y_j \notin V_i \end{cases}$$

Архитектура рассматриваемого КБО с распределенными по аппаратной платформе функциональными задачами, приведена на рисунке 2.7.

Блок-схема алгоритма приведена на рисунке 2.8.

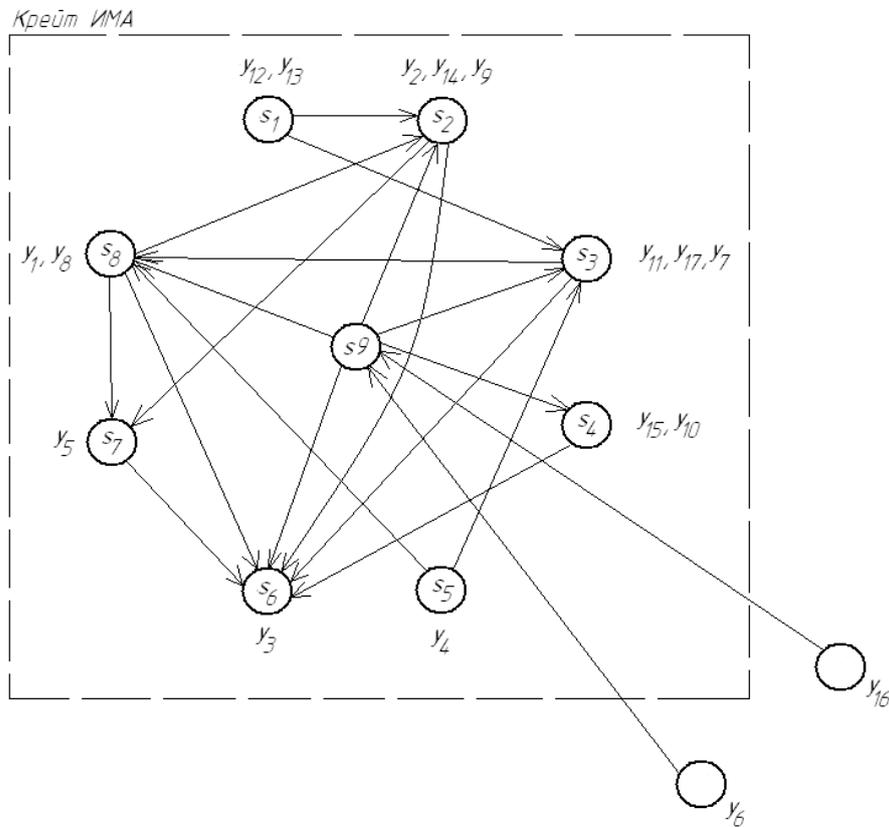


Рисунок 2.7 - Граф аппаратной структуры КБО ИМА с распределенными функциональными задачами [83]

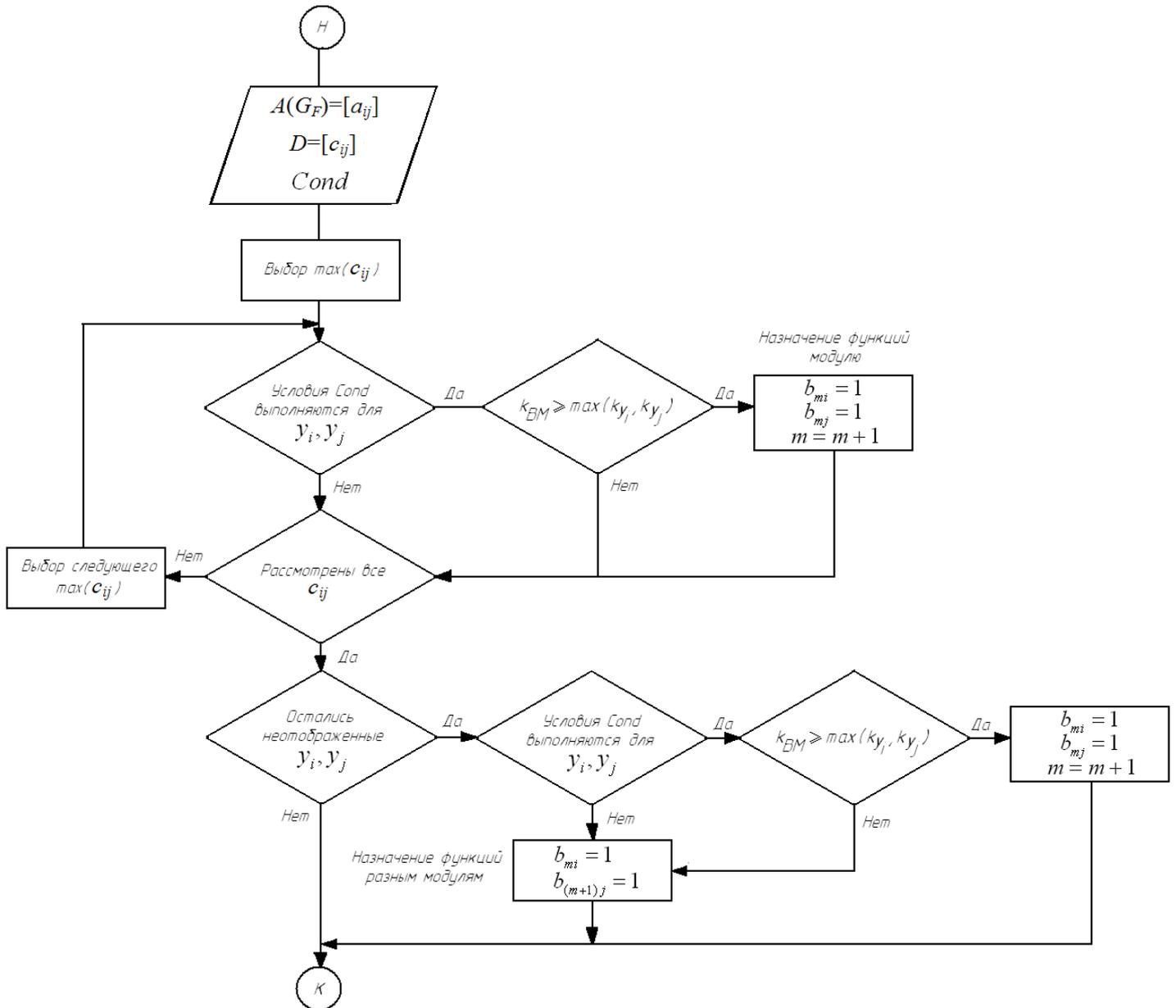


Рисунок 2.8 – Блок-схема алгоритма распределения по критерию минимума загрузки сети

При этом дуга результирующего графа G_S может представлять собой набор дуг исходного графа G_F . Происходит это в том случае, если все дуги этого набора инцидентны одной и той же паре вершин. В этом случае такой набор может быть заменен одной дугой с максимальным для них объемом передаваемого массива и максимальной частотой обращения. Такая организация позволяет снизить нагрузку на коммутатор из-за меньшего количества каналов, а также значительно упростить топологию сети.

Матрица $R(G_s)$, полученная в результате работы алгоритма, для графа аппаратной структуры, изображенной на рис. 2.7 выглядит следующим образом:

$$R(G_s) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Нулевые столбцы соответствуют функциям, выполняющимся на специализированных модулях или вне крейта ИМА.

Из рисунка 2.7 видно, что количество дуг, соединяющих вершины, сократилось. Следовательно, сократился и суммарный объем передаваемых данных по сети и задержки при обработке информации. Вершины s_1-s_5 , s_7 , s_8 представляют собой базовые ВМ, вершина s_6 – МГО, s_9 – МКС.

Предложенный алгоритм будет более эффективен при проектировании сложных систем авионики, в которых решается большое количество различных задач и используется большое число модулей. В отличие от алгоритмов решения типовых задач данный алгоритм обладает следующими преимуществами:

- возможность внесения большого числа ограничений, что не приведет к возрастанию числа вариантов перебора и времени выполнения. Но такое внесение может привести к росту числа необходимых аппаратных модулей для отображения задач на последних шагах алгоритма из-за сложности одновременного удовлетворения множества ограничений;

- отсутствие необходимости поиска максимального потока. Фактически нужно лишь перебрать все дуги, соединяющие вершины. Как следствие, добавление новых вершин и дуг ведет к линейному (а не экспоненциальному или полиномиальному) возрастанию сложности алгоритма. При использовании алгоритмов поиска максимального потока в сложных системах может оказаться невозможным отоб-

ражение соответствующих вершин на один вычислитель (из-за накладываемых ограничений);

- возможность применения на самых ранних стадиях проектирования системы, когда известны лишь набор функций, требования к обмену информацией (объем данных, скорость передачи и т.д.) и характеристики ВМ. При этом количество необходимых ВМ определяется в конце работы алгоритма и автоматически удовлетворяет всем ограничениям, накладываемым на систему (кроме стоимости).

Недостатками приведенного алгоритма являются:

- рост стоимости системы при внесении в нее множества сложных задач и соответствующем увеличении количества модулей, необходимых для их реализации;

- зависимость эффективности от целевой функции. Например, при задании в качестве критерия оптимальности минимума загрузки процессора ВМ, алгоритм будет фактически сведен к перебору всех возможных комбинаций вершин (то есть к типовому алгоритму решения задачи об упаковке).

Также следует отметить, что необходима разработка комплексного алгоритма, решающего задачу оптимального распределения задач одновременно по множеству критериев (минимальная загрузка сети передачи данных, минимальная нагрузка процессоров, минимальная нагрузка памяти и т. д.). При этом возможно разделение многомерной функциональной задачи на отдельные подзадачи, которые требуют обеспечения каких-то частных функционалов оптимальности.

Для сохранения эффективности выполнения возложенных на КБО функций при его реконфигурации распределение приложений необходимо вести именно по какому-либо оптимальному алгоритму. Это обусловлено тем фактом, что работу системы во всех режимах обеспечивает единая платформа ИМА, а это подразумевает ограниченность аппаратных ресурсов и их резервов, использующихся для реализации дополнительных функций. Разработанный алгоритм оптимального распределения функциональных приложений, в силу своей направленности именно на структурные составляющие КБО ИМА, может быть использован как основа для разработки алгоритмов реконфигурации.

2.4 Выводы

1. При моделировании КБО ИМА как объекта проектирования должны быть отражены следующие характеристики сложной системы:

- иерархическая структура;
- наличие связей между структурными элементами;
- глубокая интеграция структурных элементов и необходимость их проектирования с учетом взаимного влияния друг на друга;
- возможность функционирования в различных режимах;
- наличие множества параметров, характеризующих как отдельные структурные элементы, так и комплекс в целом.

2. В качестве критерия оптимальности при выборе конфигурации системы ИМА предложено использовать критерий минимума загрузки сети передачи данных. Размещение приложений и массивов данных в системе, обеспечивающее минимум загрузки сети, позволяет:

- уменьшить время, затрачиваемое на обмен данными между модулями по каналам связи, а, следовательно, время реализации и запаздывания прикладных функций (что является критичным фактором в некоторых приложениях);
- повысить надежность выполнения функций ввиду уменьшения времени использования и числа обращений к каналам связи;
- повысить достоверность передачи данных путем введения избыточности;
- обеспечить полное (или частичное) выполнение критичных функций в случае отказа сетевого коммутатора.

3. Разработана единая математическая модель, объединяющая аппаратную и программную составляющие системы ИМА. Модель отличается тем, что кроме связей между аппаратной и программной частями системы в ней учитываются особенности архитектуры аппаратной и организации программной частей. Модель позволяет разрабатывать алгоритмы управления конфигурацией систем ИМА с обеспечением требуемого уровня надежности системы в целом.

4. Показано, что возможность определения базовой конфигурации на основе заданного критерия оптимальности с учетом критичности функциональных приложений и требуемого уровня гарантии конструирования аппаратуры с использованием разработанной модели системы ИМА на основе решения одной из типовых задач на графах (поиск максимального потока, задача о назначениях, задача об упаковке и т.п.) сильно ограничена в силу архитектурных и структурных особенностей системы ИМА, и приводит либо к потере существенных данных о конструкции системы, либо к значительным вычислительным затратам, не гарантирующим нахождение приемлемого решения.

5. Разработан алгоритм определения базовой конфигурации системы ИМА на основе заданного критерия оптимальности с учетом критичности функциональных приложений и требуемого уровня гарантии конструирования аппаратуры. Предложенный алгоритм обладает следующими преимуществами:

- возможностью учета большого числа ограничений на систему, что не приводит к возрастанию числа вариантов перебора и времени поиска решения;
- отсутствием необходимости поиска максимального потока на графе;
- возможностью применения на самых ранних стадиях проектирования системы, когда известны лишь набор функций системы, требования к обмену информацией (объем данных, скорость передачи и т.д.) и характеристики ВМ.

6. На примере КБО вертолета, выполняющего 17 типовых полетных функций, показано применение разработанной модели и алгоритма определения базовой конфигурации системы ИМА.

3 АЛГОРИТМЫ ДИНАМИЧЕСКОЙ РЕКОНФИГУРАЦИИ КБО ИМА

При проектировании реконфигурирующегося КБО ключевой является разработка алгоритмов функционирования и средств реализации отказоустойчивого необслуживаемого комплекса, основными свойствами которого являются [38]:

- отсутствие необходимости обслуживания в межсервисные периоды с сохранением требуемого уровня безопасности;

- использование активной системы безопасности, позволяющей в случае возникновения отказов изменять его структуру для частичного или полного сохранения функций.

Построение комплекса, обладающего такими свойствами, возможно путем реализации динамической реконфигурации его алгоритмических, программных и аппаратных ресурсов, что позволит минимизировать резервирование и обеспечит максимально возможный уровень безопасности (в соответствии с Р-4761) и успешного завершения полета или его дальнейшего выполнения в случае возникновения отказов.

Наряду с разработкой математических моделей реконфигурирующегося КБО ИМА для решения указанной задачи необходимо создание алгоритмов встроенного контроля состояния системы, основанных на этих моделях, а также алгоритмов динамической реконфигурации системы в процессе ее функционирования.

Для создания требуемых алгоритмов нужно иметь полное описание взаимосвязей всех выполняемых функций, их ранжированность по критичности, необходимые для их реализации аппаратные ресурсы, которые могут быть заменены в случае реконфигурации комплекса (вычислительные модули, оперативная память и т.п.) и те ресурсы, замена которых затруднительна (датчики информации, устройства вывода и т.п.). Кроме того необходима классификация и структурирование функциональных отказов, которые будут определять тип требуемой реконфигурации программной части и оборудования.

3.1 Механизмы динамической реконфигурации

Существует три базовых категории отказов, которые свойственны платформе ИМА [7]:

- неправильное функционирование приложения или аппаратного устройства;
- неправильная реакция на действие исправного приложения или модуля, то есть неправильная интерпретация инструкции системой ИМА или отказ в предоставлении защиты от отказавших приложений;
- несвоевременное выполнение какой-либо функции системой ИМА, то есть отказ, обусловленный ошибкой в платформе, не относящийся к приложению или устройству.

Исходя из вышеперечисленных категорий отказов и необходимости выполнения различных наборов функций при изменении режима полета, были определены три базовых вида динамической реконфигурации [84, 85]:

- *аппаратная реконфигурация*. Требуется для перераспределения программных ресурсов (правильно работающих приложений) по функционирующим вычислительным модулям, если один или несколько модулей крейта вышли из строя;
- *программная реконфигурация*. Требуется для перераспределения правильно работающих приложений по функционирующим модулям крейта с учетом выбранного критерия оптимальности в случае, если какое-либо из приложений функционирует неверно или получает неверную информацию от внешних устройств (датчиков и устройств управления), которую нельзя использовать для вычислений;
- *функциональная реконфигурация*. Особый вид реконфигурации, который может потребоваться для выполнения специальной задачи, например перехода в режим, 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.).

Смешанная реконфигурация представляет собой функциональную реконфигурацию при одновременном отказе модуля/приложения.

В общем случае для обеспечения возможности реконфигурации в КБО к множеству функций Y добавляется множество функций Y_{rec} , осуществляющих контроль состояния программной и аппаратной частей комплекса и на основе его результатов принимающих решение о перераспределении ресурсов. По аналогии с выбором типа многопроцессорной системы для крейта ИМА возможны три варианта размещения функций $y \in Y_{rec}$:

1. Запуск на каждом ВМ своей копии приложения, отвечающего за контроль. При этом информация от всех $y \in Y_{rec}$ будет передаваться в модуль-носитель ОСРВ для принятия решения о необходимости реконфигурации;

2. Запуск соответствующего приложения на модуле-носителе ОСРВ. При этом оно само должно контролировать исправность всех оставляющих комплекса, для чего необходима организация передачи дополнительных данных встроенного контроля (от аппаратных средств и сервисов мониторинга копий ОСРВ на ВМ) по каналам связи;

3. Комбинированный подход, при котором приложения, отвечающие за контроль, размещаются как на ВМ, так и на модуле-носителе ОСРВ (см. рис. 3.1). Он наиболее всего подходит при проектировании КБО ИМА по следующим причинам:

- отсутствие необходимости передачи массивов данных по сети для их оценки в модуле-носителе ОСРВ. Нужно лишь передавать результаты встроенного контроля для принятия решения о реконфигурации;

- малая функциональность сервисов мониторинга ОСРВ, которая не позволяет осуществлять контроль с требуемой глубиной (как правило, не менее 0,95 в режиме расширенного контроля). Специализированные приложения можно настроить на анализ любого типа аппаратуры и ПО, в зависимости от требований ТЗ;

- возможность непосредственного управления (перезапуск в случае необходимости) и осуществления блокировки приложений при их отказе (без передачи данных о недостоверности по сети от модуля-носителя ОСРВ).

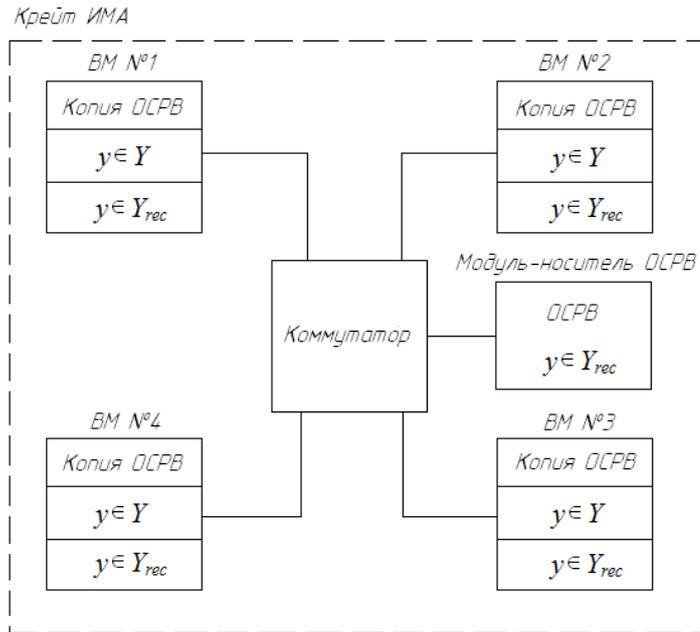


Рисунок 3.1 – Комбинированный принцип размещения функций контроля

Однако при такой реализации система приобретает дополнительные ограничения:

- необходимость учета аппаратных ресурсов, требуемых для работы приложений контроля (емкость памяти и производительность процессора):

$$\sum_{d \in D} \sigma_d x_{dv} + \sum_{y \in Y} \sigma_y x_{yv} + \sum_{y \in Y_{rec}} \sigma_y \leq L_v, \forall v \in V, \quad (3.1)$$

$$\beta_v \sum_{y \in Y} v_y \rho_y x_{yv} + \sum_{y \in Y_{rec}} \rho_y \leq R_v, \forall v \in V \quad (3.2)$$

где $\sum_{y \in Y_{rec}} \sigma_y$ - суммарный размер всех приложений контроля, размещенных в данном VM, а $\sum_{y \in Y_{rec}} \rho_y$ – требуемые для их работы ресурсы процессора VM.

- соответствующее изменение целевой функции в части ограничений при распределении задач;

- уменьшение интервала времени выполнения функциональных задач КБО. Время T_{gap} при этом разделяется на две части, одна из которых будет предназначена для работы приложений контроля;

Система контроля, изображенная на рис. 3.1, осуществляет принятие решения о необходимости реконфигурации, основываясь на данных встроенного контроля,

приходящих от множества приложений Y_{rec} и от аппаратных средств в виде разовых команд исправности. Это необходимо для более достоверного определения отказных состояний модулей, так как приложения контроля тоже могут выдавать ложные данные. При таком подходе функция модуля-носителя ОСРВ, осуществляющая реконфигурацию, по сути, является источником дополнительной опасности, которая заключается в проведении реконфигурации без необходимости. Однако, ПО авиационных комплексов разрабатывается по уровню безопасности А в соответствии с КТ-178В, и вероятность ошибки в коде, приводящей к соответствующему отказному событию, считается крайне низкой ($P = 10^{-9}$). Вероятность же отказа аппаратного обеспечения на 1-2 порядка выше, поэтому неисправная работа приложений контроля не учитывается.

Выбор того или иного типа реконфигурации основывается на нахождении системы в каком-либо состоянии, определяемом набором параметров, характеризующих программную и аппаратную части, в соответствии с выражениями (2.1)-(2.4):

$$C_S = \{C_1(S_1, Y_2), C_2(S_1, Y_2), \dots, C_f(S_1, Y_m)\},$$

$$C_H = \{C_1(S_1, Y_1), C_2(S_2, Y_2), \dots, C_n(S_k, Y_r)\},$$

$$C_T = \{C_1(S_k, Y_1), C_2(S_k, Y_2), \dots, C_p(S_k, Y_l)\},$$

$$E(C) = \left\{ \begin{array}{c} C_S \\ C_H \\ C_T \end{array} \right\}.$$

Приложение, находящееся в модуле-носителе ОСРВ и инициализирующее реконфигурацию, непрерывно формирует вектор текущего состояния системы на основе информации от встроенного контроля о количестве неисправных модулей, функциональных приложений и требуемого режима работы. При изменении какого-либо из параметров множеств S_k , Y_m , Y_r и Y_l производится сопоставление текущего состояния системы с таблицей реконфигурации, которая формируется на этапе разработки при анализе функциональных отказов и хранится в памяти модуля-носителя ОСРВ. Фактически, разработчику необходимо определить полный

перечень функциональных отказов и соответствующий ему перечень вариантов распределения ресурсов системы по какому-либо оптимальному алгоритму. Сложность этой задачи увеличивается с ростом размерности системы. Однако разработчик может принять решение об осуществлении реконфигурации только при определенном неполном наборе отказов (например, только при отказах аппаратной части или изменении режима полета).

Обобщенный алгоритм принятия решения о проведении реконфигурации состоит из следующих шагов:

Шаг 1. На основании данных встроенного контроля формируется вектор с булевыми координатами, отражающими изменения в структуре комплекса:

$$\overrightarrow{REC} = \left\{ \left\{ \begin{array}{l} 1, S_{ki} \neq S_{ki-1} \\ 0, S_{ki} = S_{ki-1} \end{array} \right\}, \left\{ \begin{array}{l} 1, Y_{ki} \neq Y_{ki-1} \\ 0, Y_{ki} = Y_{ki-1} \end{array} \right\}, \left\{ \begin{array}{l} 1, q_i \neq q_{i-1} \\ 0, q_i = q_{i-1} \end{array} \right\} \right\}, \quad (3.3)$$

где S_k – множество исправных КФМ, Y_k – множество исправных приложений из всего множества приложений Y , q – заданный режим работы. При этом индексом $i-1$ обозначается предыдущая итерация выполнения алгоритма, а i – текущая. Единица означает, что по сравнению с предыдущей итерацией произошли изменения в соответствующем составе или режиме.

Шаг 2. В зависимости от координат вектора принимается решение об изменении состояния комплекса и проведении реконфигурации:

{0,0,0}: изменений состояния комплекса не произошло, реконфигурация не требуется, переход к шагу 1;

{1,0,0}: произошел аппаратный отказ, переход от конфигурации C_{Hi} к конфигурации C_{Hi+1} , требуется аппаратная реконфигурация, переход к шагу 3;

{0,1,0}: произошел программный отказ, переход от конфигурации C_{Si} к конфигурации C_{Si+1} , требуется аппаратная реконфигурация, переход к шагу 3;

{1,1,0}: произошли аппаратный и программный отказы, переход от конфигураций C_{Si} и C_{Hi} к конфигурациям C_{Si+1} и C_{Hi+1} , требуется аппаратная реконфигурация, переход к шагу 3;

{0,0,1}: изменений состояния комплекса не произошло, смена режима работы, переход от конфигурации C_{Ti} к конфигурации C_{Ti+1} , требуется функциональная реконфигурация, переход к шагу 3;

{1,0,1}: произошел аппаратный отказ, смена режима работы, переход от конфигураций C_{Hi} и C_{Ti} к конфигурациям C_{Hi+1} и C_{Ti+1} , требуется смешанная реконфигурация, переход к шагу 3;

{0,1,1}: произошел программный отказ, смена режима работы, переход от конфигураций C_{Si} и C_{Ti} к конфигурациям C_{Si+1} и C_{Ti+1} , требуется смешанная реконфигурация, переход к шагу 3;

{1,1,1}: произошли программный и аппаратный отказы, смена режима работы, переход от конфигураций C_{Si} , C_{Hi} и C_{Ti} к конфигурациям C_{Si+1} , C_{Hi+1} и C_{Ti+1} , требуется смешанная реконфигурация, переход к шагу 3;

Шаг 3. Реализуется алгоритм выбранного на шаге 2 вида реконфигурации. Производится обнуление координат вектора REC . В случае успешного проведения реконфигурации переходят к шагу 1. В случае неудачи – к шагу 4.

Шаг 4. Реконфигурация невозможна по причине недостаточности аппаратных ресурсов или отказа функции с высоким уровнем критичности. Выдача сигнала неисправности КБО. Конец вычислений.

Шаги 1-3 выполняются циклически на протяжении всего времени функционирования системы.

Решение о выдаче сигнала неисправности может быть принято уже на шаге 1 при анализе входных данных об отказах. Например, при выходе из строя слишком большого количества модулей не имеет смысла осуществлять попытки реконфигурации. Для реализации этого подхода, помимо анализа вектора REC , необходимо проверять заданные ограничения по количеству отказов. Кроме того, в перечень возможных состояний комплекса можно добавить такие варианты, при которых будет осуществляться проверка возможности выполнения функций с высоким уровнем критичности без оптимального их распределения. Это может оказаться полезным в сложных системах с большим количеством модулей, в которых

существует возможность такого распределения с целью сохранения ее работоспособности при множественных отказах.

Блок-схема алгоритма принятия решения о реконфигурации приведена на рисунке 3.2.

Переход между конфигурациями i и $i+1$ подразумевает обращение к таблице реконфигурации и выбора необходимого состояния КБО $E(C)$ на основе множеств S_k , Y_m , Y_r и Y_l , формирование которых происходит каждый раз после принятия решения о необходимости реконфигурации, и которые являются входными данными для соответствующих алгоритмов. При этом:

- множество S_k состоит из всех возможных комбинаций отказов модулей, которые допускают выполнение функций с высокими уровнями критичности (даже без учета оптимального распределения);

- множество Y_m состоит из всех программных приложений, соответствующих текущему режиму за вычетом отказавших функций;

- множество Y_r состоит из всех программных приложений, соответствующих текущему режиму за вычетом функций, исключенных из состава КБО по причине недостатка аппаратных ресурсов;

- множество Y_l состоит из всех программных приложений для данного режима.

С учетом этого, отношения множеств Y_m , Y_r , Y_l и Y выглядят следующим образом:

$$\bigcup_l Y_l = Y, Y_m \subset Y_l, Y_r \subset Y_l, Y_m \cup Y_r = Y_l.$$

То есть существуют функции, общие для всех режимов работы. При этом отказы могут влиять как на общие функции, так и на присущие лишь отдельным режимам. Выделение отдельных множеств Y_m и Y_r обусловлено необходимостью определения точного характера отказа для правильного принятия решения о типе необходимой реконфигурации.

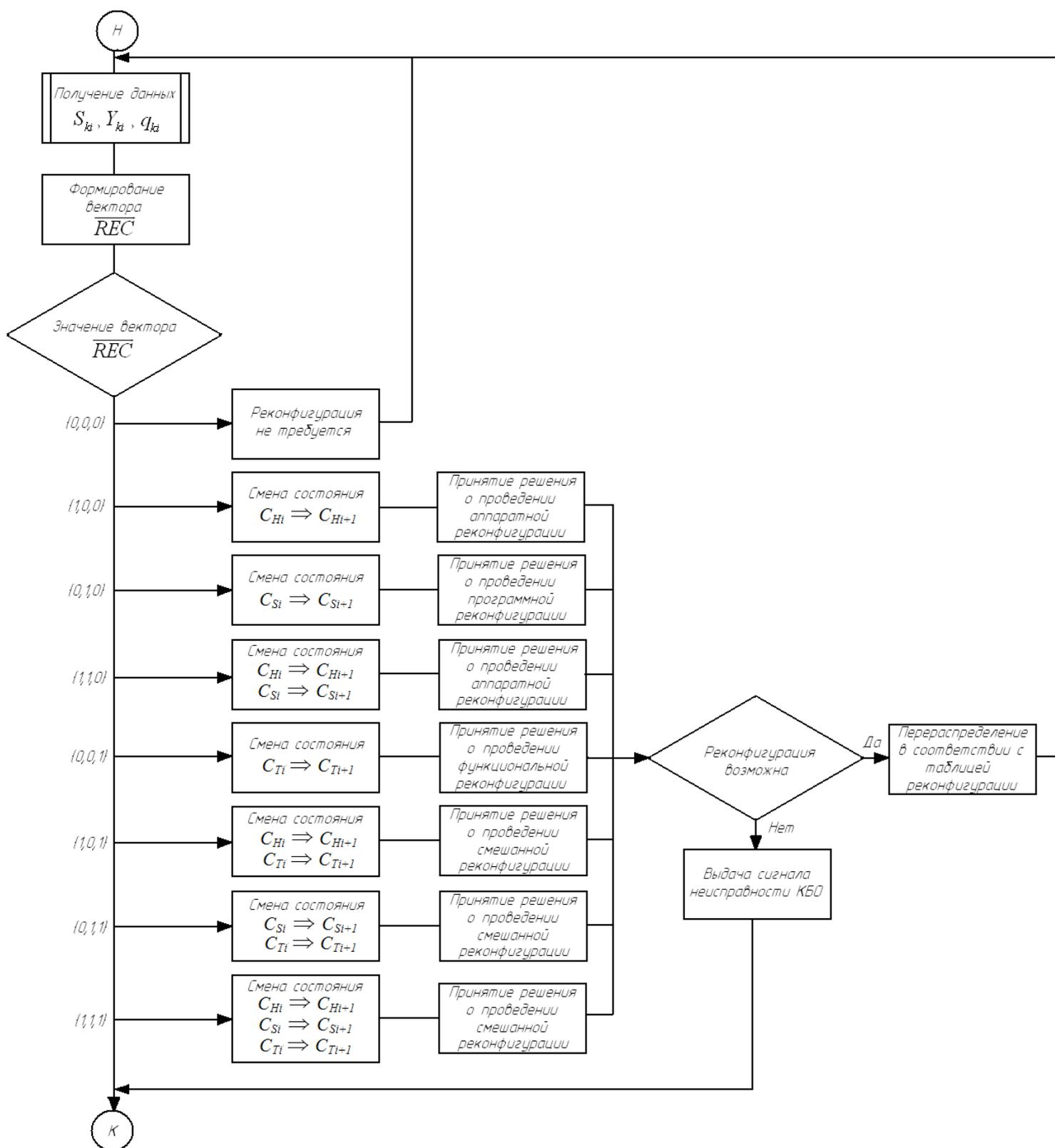


Рисунок 3.2 – Блок-схема алгоритма принятия решения о реконфигурации

Таким образом, обобщенный процесс реконфигурации заключается в переходе между различными состояниями комплекса по многомерной суперповерхности (см. рис. 2.2) при возникновении отказов или изменении режима работы с последующим перераспределением ресурсов на основе данных таблицы реконфигурации:

$$E_i(C) = \{C_{S_i}, C_{H_i}, C_{T_i}\} \Rightarrow E_{i+1}(C) = \{C_{S_{i+1}}, C_{H_{i+1}}, C_{T_{i+1}}\}.$$

В объективной форме таблица реконфигурации представляет собой базу данных, систематизированную на основе наборов множеств S_k , Y_m , Y_r и Y_l и содержащую информацию о необходимом распределении приложений по модулям. Данное распределение может быть как назначенным вручную, так и полученным на основе какого-либо алгоритма. Содержание и полнота этой базы данных определяется разработчиком на основании требований ТЗ к перечню функций, режимам работы и устойчивости к отказам. Поскольку требования к системе могут быть различными, то необходимо разработать алгоритмы перераспределения ресурсов для всех типов реконфигурации в отдельности.

3.2 Алгоритм аппаратной динамической реконфигурации

Пусть S_k – множество отказавших модулей крейта. При этом аппаратная платформа КБО представляется ориентированным графом $G_S=(V, W)$, где вершины $v \in V$ соответствуют модулям, а ребра $w \in W$ – каналам связи между ними в соответствии с математическим описанием структуры КБО ИМА.

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 и ВМ $k \in K$ и множество отказавших модулей S_k . Функциональный граф подсистемы КБО ИМА с назначенными уровнями критичности представлен на рисунке 3.3.

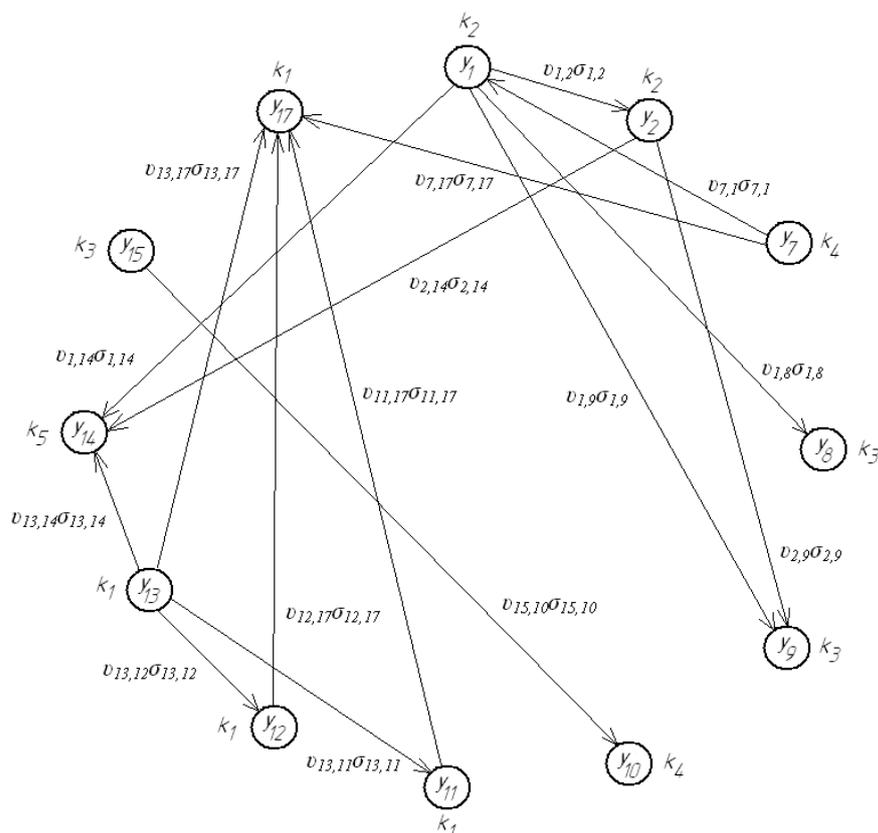


Рисунок 3.3 – Функциональный граф подсистемы КБО вертолета с назначенными параметрами критичности

КБО, как правило, проектируются с учетом необходимости наличия неких резервов (аппаратных и программных), которые задействуются в случае какого-

либо отказа. Однако этот резерв чаще всего не является автономным и теряется при отказе крупного элемента структуры (например, резерв вычислительной мощности модуля бесполезен при полном выходе из строя этого модуля). Поэтому ситуация, в которой после отказа возможно будет перераспределить абсолютно все функции по исправно работающим аппаратным ресурсам, маловероятна. Более того, даже если такая ситуация и произойдет, то нет гарантии того, что оставшиеся ресурсы позволят перераспределить функции по заданному критерию оптимальности.

Перед началом работы алгоритма реконфигурации функции $y \in Y$, необходимо упорядочить таким образом, чтобы впоследствии их было удобно исключать из процесса распределения по аппаратной платформе. При упорядочении используются параметры критичности k и суммарный выходной поток данных $v_m \sigma_m$, идущий между связанной парой функций.

Функции должны располагаться в порядке убывания параметра k и убывания параметра $v_m \sigma_m$. Дополнительные ограничения на возможность исключения накладывает информационная связанность, при которой выходные данные одной функции служат входными для другой, не связанной с ней максимальным потоком данных. При реализации какого-либо алгоритма сортировки на каждой его итерации необходимо проверять именно цепочку функций, потерянных в результате исключения. Особое внимание следует обращать на возможные потери функций с уровнями критичности k_1 и k_2 , что недопустимо с точки зрения сохранения функционирования КБО после проведения реконфигурации.

Для КБО, изображенного на рисунке 3.3, функции разделяются так, как показано в таблице 3.1. При этом полагается, что выполняется условие (2.13):

$$\begin{aligned} v_{1,2}\sigma_{1,2} > v_{13,12}\sigma_{13,12} > v_{1,8}\sigma_{1,8} > v_{1,9}\sigma_{1,9} > v_{1,14}\sigma_{1,14} > v_{13,11}\sigma_{13,11} > v_{2,14}\sigma_{2,14} > \\ v_{15,10}\sigma_{15,10} > v_{2,9}\sigma_{2,9} > v_{12,17}\sigma_{12,17} > v_{11,17}\sigma_{11,17} > v_{13,14}\sigma_{13,14} > \\ > v_{13,17}\sigma_{13,17} > v_{7,1}\sigma_{7,1} > v_{7,17}\sigma_{7,17}. \end{aligned}$$

Таблица 3.1 – Упорядоченное множество функций КБО

$k \in K$	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					

Перераспределение функций в случае отказа по исправной аппаратной платформе осуществляется следующим образом.

Шаг 1. Проверяется условие

$$\beta_v \sum_{y \in Y} v_y \rho_y \leq \sum_{v \in V} R_v, \forall v \in V, \forall y \in Y$$

где β_v – коэффициент запаса на основе экспертных оценок

Если условие выполняется, то КБО имеет достаточное количество процессорных ресурсов для выполнения всех функций, не распределенных по какому-либо критерию оптимальности. Переход к шагу 2. Если условие не выполняется, то становится очевидной необходимость исключения каких-либо функций из состава комплекса. Шаг 2 пропускается и осуществляется переход к шагу 3.

Шаг 2. Реализуется алгоритм распределения функций по какому-либо заранее заданному критерию оптимальности (например, по минимуму загрузки сети [83]). В случае его успешной реализации реконфигурация комплекса на этом завершается, поскольку все функции заново назначены для работы в исправных модулях. Если реализовать оптимальный алгоритм распределения по какой-то причине не удалось (отсутствует необходимое число дополнительных модулей для назначения оставшихся в ходе распределения задач, либо нарушается ход алгоритма при “ручном” назначении задачи в какой-либо модуль), то либо распределяют функции неоптимальным образом, либо переходят к шагу 3.

Шаг 3. Из всего множества функций $y \in Y$ выбирается функция с наименьшими значениями уровня критичности и наименьшим суммарным потоком данных от этой функции и исключается из состава КБО. При этом ее исключение должно

приводить к дополнительному исключению минимального числа других, не связанных с ней максимальным потоком, функций. Затем реализуется алгоритм распределения функций по какому-либо заранее заданному критерию оптимальности. В случае его успешной реализации реконфигурация комплекса на этом завершается. Если реализовать алгоритм не удалось, то переходят к шагу 4.

Шаг 4. Выбирается функция с тем же значением критичности k , что и в шаге 3, но с большим по сравнению с ней выходным суммарным потоком $v_m\sigma_m$ (но меньшим, чем у всех остальных) или, если функции с данным уровнем критичности отсутствуют, то выбирается функция со следующим уровнем и с минимальным для данного уровня суммарным выходным потоком данных и минимальной цепочкой дополнительно исключаемых функций (для G_F порядок исключения будет следующим: 14, 10, 10-14, 5, 5-14, 5-10-14 и т.д.).

Данный порядок исключения функций основан, в том числе, и на том, что необходимо минимизировать потери данных, которые возникнут при удалении функции из состава КБО. Это делается для того, чтобы оставшиеся функции работали максимально эффективно после реконфигурации в условиях аппаратного отказа и недостатка входных данных. Ведь если функция потеряет весь поток данных, который ей необходим, то она не сможет выполняться дальше.

Если распределение не удалось, и были выбраны все функции со всеми возможными допустимыми уровнями критичности, то распределение функциональных задач по заданному критерию оптимальности для данной отказной ситуации невозможно, и нужно либо менять критерий оптимальности, либо осуществлять неоптимальное распределение, либо выставлять сигнал полной неисправности КБО.

Шаг 4 повторяется до тех пор, пока не будут исключены все функции с отличными от заданного уровнями критичности. **Конец вычислений.**

Необходимо отметить, что для разработчика КБО изначально определены задачи с необходимым уровнем критичности k (в соответствии с КТ-178В и КТ-254), исключение которых из комплекса недопустимо из-за невозможности

дальнейшего выполнения полетного задания. Как правило, это уровни k_1 и k_2 (см. табл. 3.1).

Блок-схема алгоритма приведена на рисунке 3.4. Результат его работы в КБО, изображенном на рисунке 3.3, представлен на рисунках 3.5(а) и 3.5(б) при отказе, соответственно, одного и двух аппаратных модулей.

Возможность осуществления процедуры неоптимального распределения может проверяться после каждого исключения какой-либо функции из состава КБО. Это будет полезно в тех случаях, когда сохранение как можно большего числа рабочих приложений предпочтительнее их оптимального размещения (например, при потере данных, необходимых для выполнения критичной функции, после исключения).

Критерий оптимальности и упорядоченное множество функций КБО также могут быть различными и учитывать специфику проектируемой системы и требования ТЗ в части отказоустойчивости.

Таким образом, аппаратная динамическая реконфигурация может использоваться для парирования лишь аппаратных отказов в системах, в которых отсутствует резервирование, но к которым, тем не менее, предъявляются повышенные требования к живучести. Для обеспечения функциональной безопасности КБО ИМА предпочтительнее использовать программную или смешанную динамические реконфигурации.

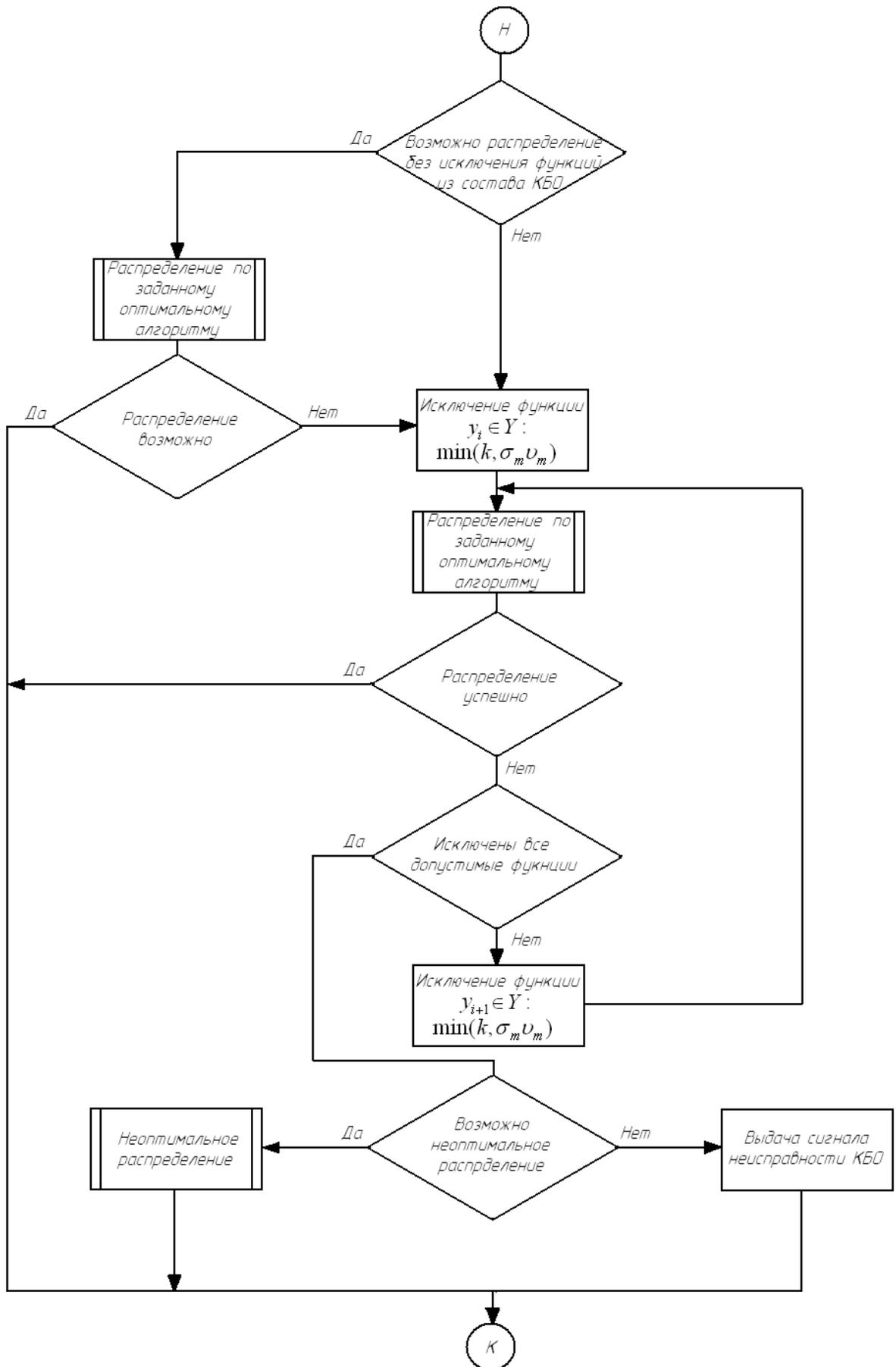
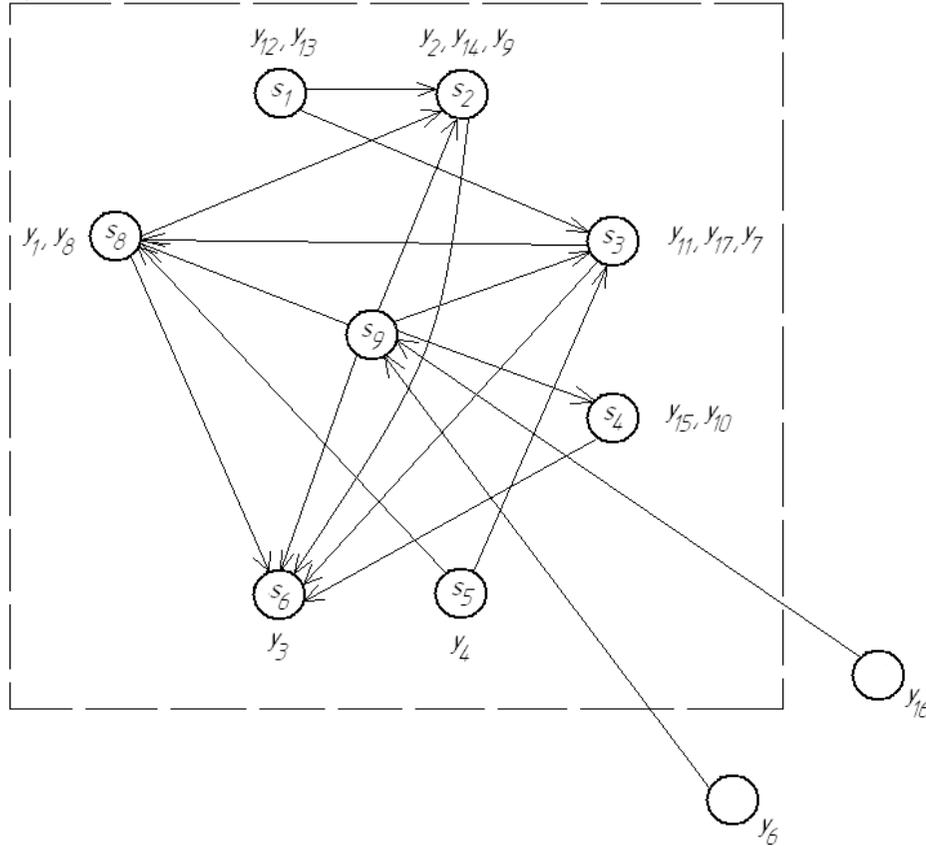


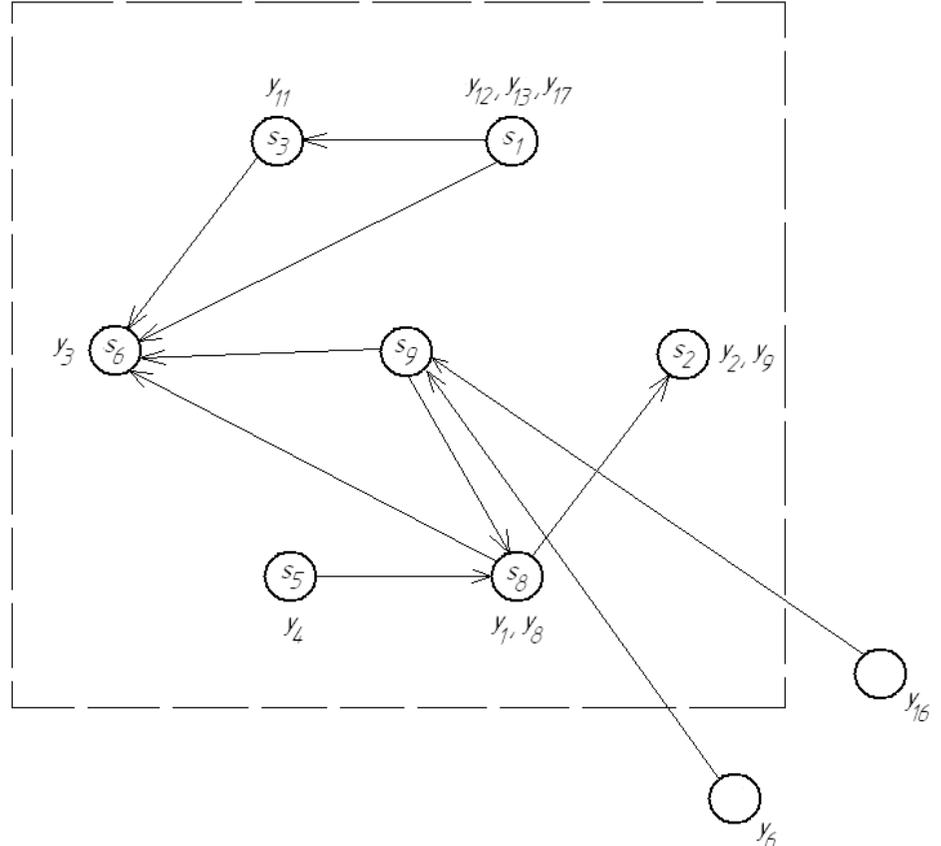
Рисунок 3.4 – Блок-схема алгоритма аппаратной реконфигурации КБО ИМА

Крейт ИМА



б) Отказал 1 модуль. исключена задача 5

Крейт ИМА



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

Рисунок 3.5 – Граф G_S КБО ИМА вертолета после работы алгоритма аппаратной реконфигурации

3.3 Программная и смешанная реконфигурации.

Алгоритм смешанной динамической реконфигурации

Программная реконфигурация, также как и аппаратная, подразумевает прямую необходимость перераспределения функциональных приложений по аппаратной платформе при возникновении отказов. 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:

$$\min \left\{ \sum_{m \in Y} \sum_{n \in Y} v_{n,m} \sigma_{n,m} \right\} \leq \min \left\{ \sum_{m \in Y_r} \sum_{n \in Y_r} v_{n,m} \sigma_{n,m} \right\}, \quad (3.4)$$

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 очевидно, что повторное применение алгоритма распределения по алгоритму минимума загрузки сети передачи данных может привести к еще большему уменьшению нагрузки на сеть. Это достигается за счет возникновения разрыва некоторых единичных связей между приложениями (отказавшими и рабочими), которые имели большее значение суммарного потока данных, что позволяет определить их в модули с другими рабочими приложениями, связь с которыми не нарушается [84].

Также дело обстоит, если используется другой критерий, например, критерий минимума загрузки процессора каждого ВМ. В случае отказа какого-либо приложения в системе освобождаются дополнительные процессорные ресурсы, что обязательно вызывает необходимость перераспределения для выполнения условий оптимальности.

Однако иногда выигрыш от программной реконфигурации оказывается незначительным, а затраты времени и ресурсов для ее проведения – существенными. Поэтому разработчик системы в системных алгоритмах и требованиях к ПО должен заранее определить набор функций, отказ которых будет приводить к перераспределению задач. Это очень трудоемкий процесс, поскольку количество различных вариаций отказов растет в геометрической прогрессии с ростом количества приложений. Следовательно, 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.

В общем случае, программная реконфигурация проводится при неизменном числе аппаратных отказов S_k и заключается в переходе между состояниями $\{C_1(S_1, Y_2), C_2(S_1, Y_2), \dots, C_f(S_1, Y_m)\}$ при выходе из строя какого-либо приложения, где S_1 – набор неисправных модулей в данный момент времени. Основным отличием данных состояний от состояний $C_H = \{C_1(S_1, Y_1), C_2(S_2, Y_2), \dots, C_n(S_k, Y_r)\}$ заключается в том, что набор функций во множествах Y_m и Y_r изменяется в результате исключения их из состава КБО (по причине выхода из строя модулей) и в результате программных отказов соответственно. При этом нужно учитывать, что средствами ОСРВ возможно изолировать и закрыть неисправное по причине внутренних скрытых ошибок (например, некорректные входные данные, приводящие к исключительной ситуации - exception) программное приложение, запустив точную работоспособную копию. Программная реконфигурация рассматривает лишь такие отказы приложений, которые не могут быть парированы, например отказ внешней системы или датчика, информация от которых является входной для данного приложения.

Фактически, задача программной реконфигурации идентична задаче распределения приложений и заключается в минимизации (максимизации) целевой функции с учетом ограничений. При этом из графа функциональной структуры G_F ис-

ключаются вершины и инцидентные дуги, соответствующие отказавшим приложениям. Соответствующие изменения вносятся в матрицы $A(G_F)$ и $D(G_F)$.

Алгоритм программной реконфигурации также повторяет алгоритм оптимального распределения ресурсов, а процесс ее осуществления заключается в переходе между следующими состояниями комплекса:

$$E_i(C) = \{C_{S_i}, C_{H_i}, C_{T_i}\} \Rightarrow E_{i+1}(C) = \{C_{S_{i+1}}, C_{H_i}, C_{T_i}\}.$$

То есть он направлен на оптимальное перераспределение приложений по множеству модулей (достаточному для переназначения без исключения функций из состава КБО) при неизменном режиме полета.

Функциональная реконфигурация представляет собой особый вид реконфигурации, который может потребоваться для выполнения специальной задачи, например перехода в режим, 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.).

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

Основным отличием функциональной реконфигурации от реконфигураций других типов является определенность конечных состояний КБО, то есть 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 критерием. 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 соста-

вить алгоритмы физического переключения аппаратной платформы в процессе реконфигурации при переходах между режимами [84].

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 реконфигураций.

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

Для выполнения различных задач в полете определено множество режимов Q . Количество режимов $q \in Q$, задается в техническом задании и заранее известно. Перечень функций $y \in Y_l$, выполняющихся в режиме $q \in Q$, также определен и представляет собой ориентированный граф функциональной структуры $G_{Fq}=(Y_l, Z_q)$, где $z \in Z_q$ – дуга, соответствующая информационной связи между двумя функциями из множества Y_l .

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

$$\bigcup_q Y_{l_q} = Y. \quad (3.5)$$

Соответственно объединение графов, соответствующих режимам полета, даст нам исходный граф функциональной структуры G_F :

$$\bigcup_q G_{Fq} = G_F. \quad (3.6)$$

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 перераспределение должно учитывать критич-

ность функций $k \in K$, количество отказавших модулей $m \in S_k$ и приложений $y \in N_y$, а также установленный режим полета $q \in Q$.

В дальнейшем полагается, что КБО должен функционировать в двух режимах – режим крейсерского полета и режим боя. Для этого добавляются следующие дополнительные задачи:

18. Управление вооружением;
19. Управление системами постановки помех;
20. Многоканальная система наведения.

Функциональные графы G_{F1} и G_{F2} данного КБО для крейсерского полета и боевого режима соответственно представлены на рисунке 3.6. При этом задачи под номерами 18-20 характерны только графа G_{F2} , а задачи 8-10 – только для графа G_{F1} .

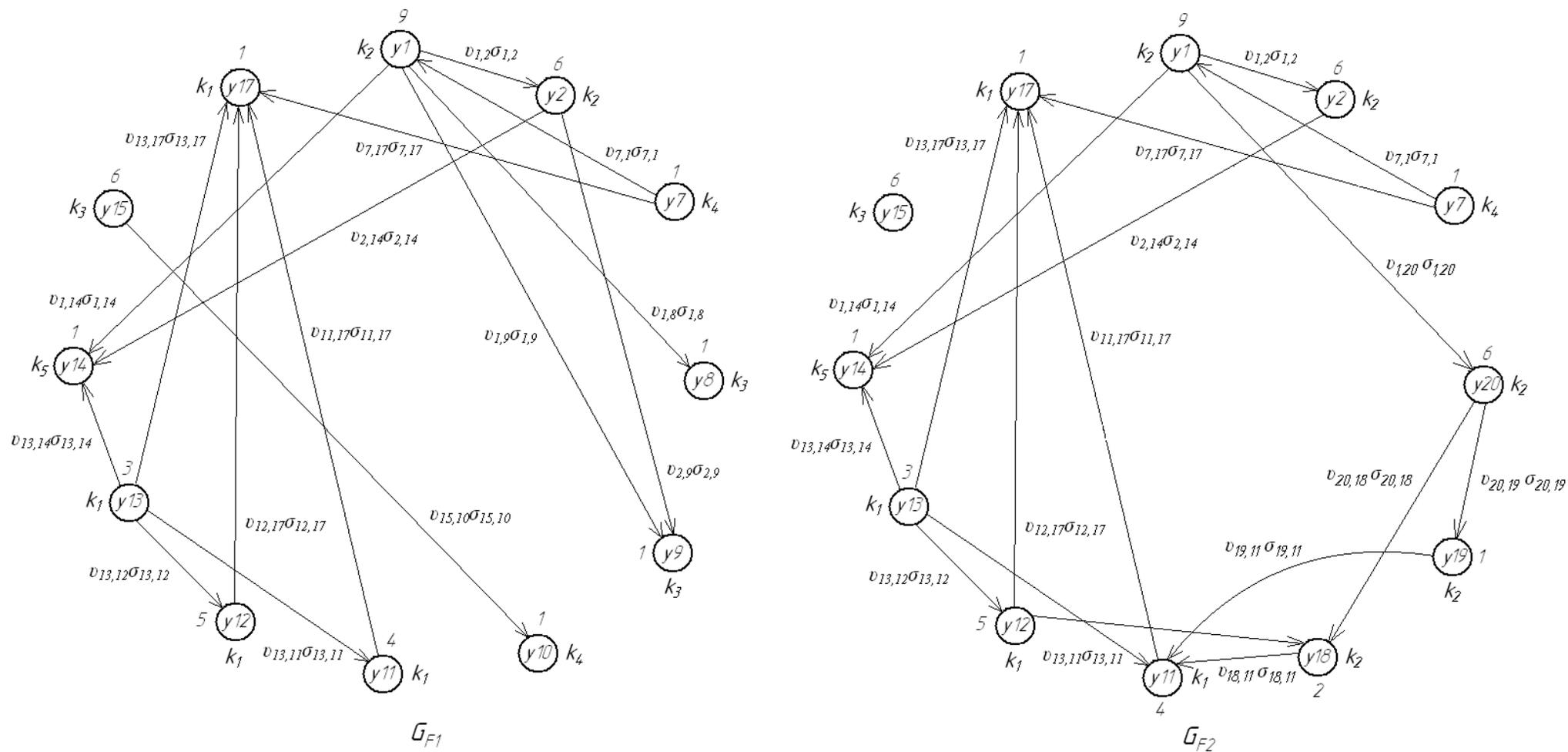


Рисунок 3.6 – Функциональные графы режимов работы КБО

Перед началом работы алгоритма реконфигурации функции $y \in Y$ необходимо упорядочить таким же образом, как и при аппаратной реконфигурации, то есть в порядке убывания параметра k , убывания параметра $v_m \sigma_m$ и с минимальными цепочками исключаемых функций. Для рассматриваемого КБО функции разделяются так, как показано в таблицах 3.2 и 3.3. При этом полагается, что:

$$\begin{aligned}
 &v_{1,2}\sigma_{1,2} > v_{13,12}\sigma_{13,12} > v_{1,8}\sigma_{1,8} > v_{1,9}\sigma_{1,9} > v_{1,14}\sigma_{1,14} > v_{13,11}\sigma_{13,11} > v_{2,14}\sigma_{2,14} > \\
 &> v_{15,10}\sigma_{15,10} > v_{2,9}\sigma_{2,9} > v_{12,17}\sigma_{12,17} > v_{1,20}\sigma_{1,20} > v_{11,17}\sigma_{11,17} > v_{13,14}\sigma_{13,14} > \\
 &> v_{13,17}\sigma_{13,17} > v_{20,18}\sigma_{20,18} > v_{20,19}\sigma_{20,19} > v_{19,11}\sigma_{19,11} > v_{18,11}\sigma_{18,11} > \\
 &> v_{12,18}\sigma_{12,18} > v_{7,1}\sigma_{7,1} > v_{7,17}\sigma_{7,17}.
 \end{aligned} \tag{3.7}$$

Таблица 3.2 – Упорядоченное множество функций КБО для режима крейсерского полета

$k \in K$	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					

Таблица 3.3 – Упорядоченное множество функций КБО для режима боя

$k \in K$	Функции $y \in Y$					
k_1	13	11	12	17	4	3
k_2	2	1	18	19	20	
k_3	15					
k_4	5	7				
k_5	14					

Перераспределение функций в случае отказа по исправной аппаратной платформе осуществляется следующим образом [84].

Шаг 1. Определяется заданный режим работы КБО q , количество отказавших ВМ S_k и приложений $y \in N_y$. Переход к шагу 2.

Шаг 2. Проверяется условие для заданного режима полета $q \in Q$

$$\beta_v \sum_{y \in Y} v_y \rho_y \leq \sum_{v \in V} R_v, \forall v \in (V - S_k), \forall y \in (Y - N_y) \quad (3.6)$$

где β_v – коэффициент запаса на основе экспертных оценок

Если условие выполняется, то КБО имеет достаточное количество исправных аппаратных ресурсов для выполнения всех исправных функций, не распределенных по какому-либо критерию оптимальности. Переход к шагу 3. Если условие не выполняется, то становится очевидной необходимость исключения каких-либо функций из состава комплекса. Шаг 3 пропускается и осуществляется переход к шагу 4.

Шаг 3. Реализуется алгоритм распределения функций по какому-либо критерию оптимальности (например, по минимуму загрузки сети). В случае его успешной реализации реконфигурация комплекса на этом завершается, поскольку все исправные функции заново назначены для работы в исправных модулях. Если реализовать алгоритм распределения по какой-то причине не удалось (отсутствует необходимое число дополнительных модулей для назначения оставшихся в ходе распределения задач, либо нарушается ход алгоритма при “ручном” назначении задачи в какой-либо модуль), то переходят к шагу 4.

Шаг 4. Из всего множества функций $y \in Y$ выбирается функция с наименьшими значениями уровня критичности и наименьшим суммарным потоком данных от этой функции и исключается из состава КБО. При этом ее исключение должно приводить к дополнительному исключению минимального числа других, не связанных с ней максимальным потоком, функций. Затем реализуется алгоритм распределения функций по какому-либо заранее заданному критерию оптимальности. В случае его успешной реализации реконфигурация комплекса на этом завершается. Если реализовать алгоритм не удалось, то переходят к шагу 5.

Шаг 5. Выбирается функция с тем же значением критичности k , что и в шаге 4, но с большим по сравнению с ней выходным суммарным потоком $v_m \sigma_m$ (но меньшим, чем у всех остальных) или, если функции с данным уровнем критичности отсутствуют, то выбирается функция со следующим уровнем и с минимальным для данного уровня суммарным выходным потоком данных и минимальной цепочкой дополнительно исключаемых функций (для режима боя порядок исключения будет следующим: 14, 5, 5-14, 7, 7-14, 7-5, 7-5-14 и т.д.).

Если распределение не удалось и были выбраны все функции со всеми возможными допустимыми уровнями критичности, то распределение функциональных задач по заданному критерию оптимальности для данной отказной ситуации невозможно, и нужно либо менять критерий оптимальности, либо осуществлять неоптимальное распределение, либо выставлять сигнал полной неисправности КБО.

Шаг 5 повторяется до тех пор, пока не будут исключены все функции с отличным от заданного уровнями критичности. **Конец вычислений.**

Данный алгоритм представляет собой результат объединения алгоритмов решения задачи оптимального распределения функций КБО и аппаратной реконфигурации при наличии нескольких режимов работы. Для его выполнения в вычислителе необходима корректировка входных данных:

- матрицу смежности $A(G_F)$ необходимо разбить на q матриц $A_q(G_{Fq})$ в соответствии с режимами работы системы;
- матрицу потоков данных $D(G_F)$ необходимо разбить на q матриц $D_q(G_{Fq})$ в соответствии со связями в матрицах смежности;
- набор ограничений, в общем случае, корректируется исходя из особенностей установленного режима и соответствующих требований к аппаратным ресурсам. То есть для каждого $q \in Q$ определяется свой набор ограничений в виде системы неравенств $Cond_q$.

Выходными данными алгоритма служат q прямоугольных матриц $R_q(G_s)$, построенных на основе графа аппаратной структуры для каждого из режимов работы.

Блок-схема алгоритма приведена на рисунке 3.7.

Результат работы данного алгоритма над КБО, функциональные графы режимов которого представлены на рисунке 3.6, представлен на рисунках 3.8 а) и 3.8 б) при отказе двух аппаратных модулей в режиме крейсерского полета и в режиме боя соответственно. Как видно из рисунка, все функции с уровнями критичности k_1 и k_2 отображены на вершины графа аппаратной структуры G_S .

Следует также отметить, что в случае отказа ВМ или приложения в каком-либо режиме простое переключение на другой режим с сохранением оптимального распределения функциональных задач невозможно, поскольку предыдущий процесс реконфигурации (если он был) эти отказы не учитывал. Поэтому для эффективной работы КБО необходимо производить расчеты по перераспределению функций для всех режимов и сохранять их в виде таблицы реконфигурации для последующего использования при переключении. Это позволит избежать опасных ситуаций, когда переход в другой режим невозможен, и, в случае необходимости, функционировать в аварийном режиме для совершения экстренной посадки.

При проектировании сложной системы ИМА в соответствии с Р-297 рекомендуется использовать именно комбинированный принцип размещения функций контроля в сочетании с алгоритмами смешанной реконфигурации и принятия решения о ее проведении, так как они удовлетворяют всем требованиям стандарта по мониторингу рабочего состояния и управления неисправностями (п. 3.6 Р-297):

1. Обладают средствами для определения типа отказа как аппаратной платформы, так и программных приложений.
2. Задают правила и последовательность назначения приложений при возникновении отказов.
3. Учитывают перечни минимально необходимого бортового оборудования и функций с высоким уровнем критичности.
4. Определяют ответные меры на каждый тип отказа.

Для определения уровня надежности и безопасности реконфигурирующегося КБО и сравнения его с комплексом традиционной федеративной архитектуры бы-

ла проведена соответствующая оценка с использованием автоматизированной системы расчета.

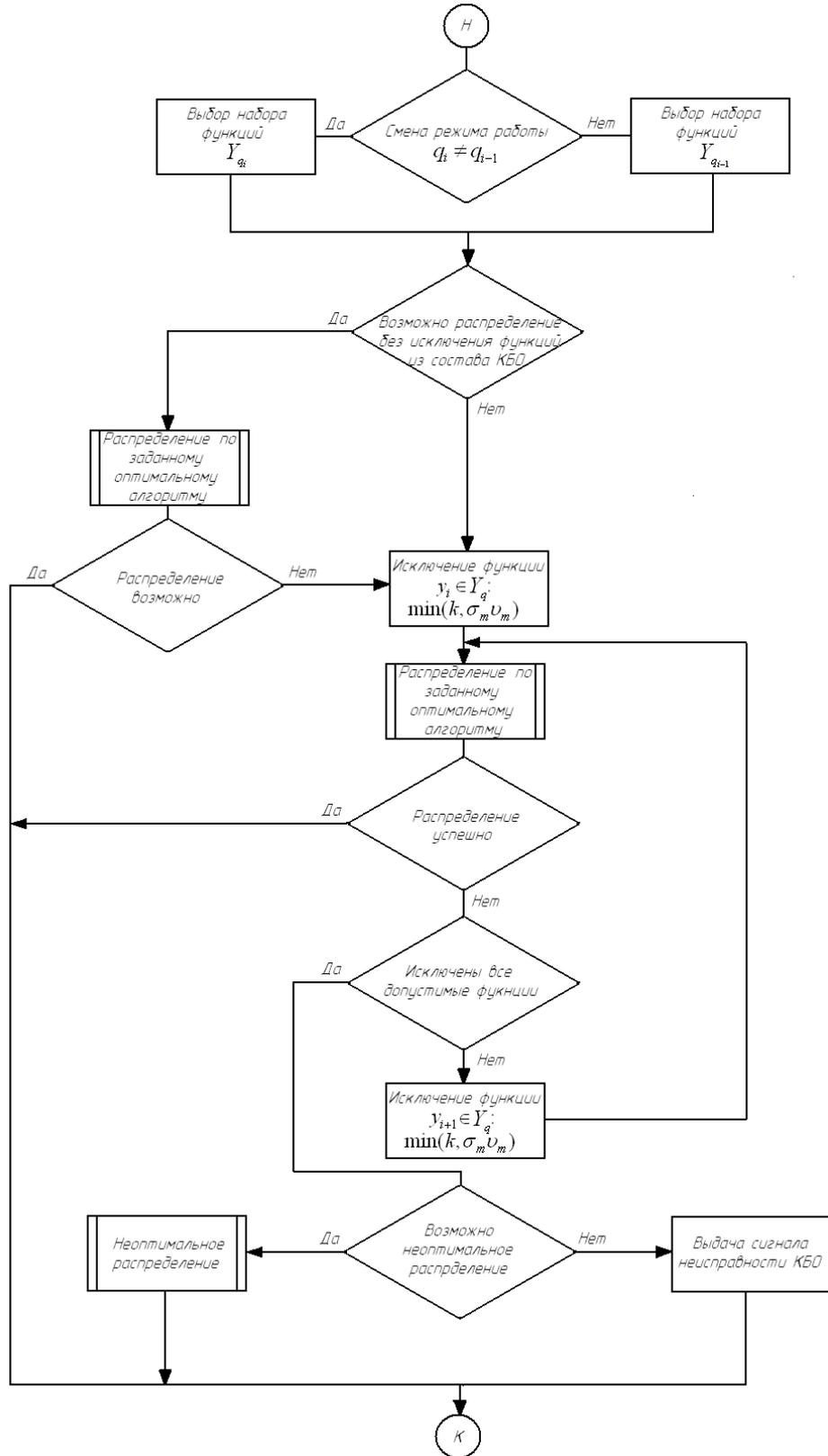
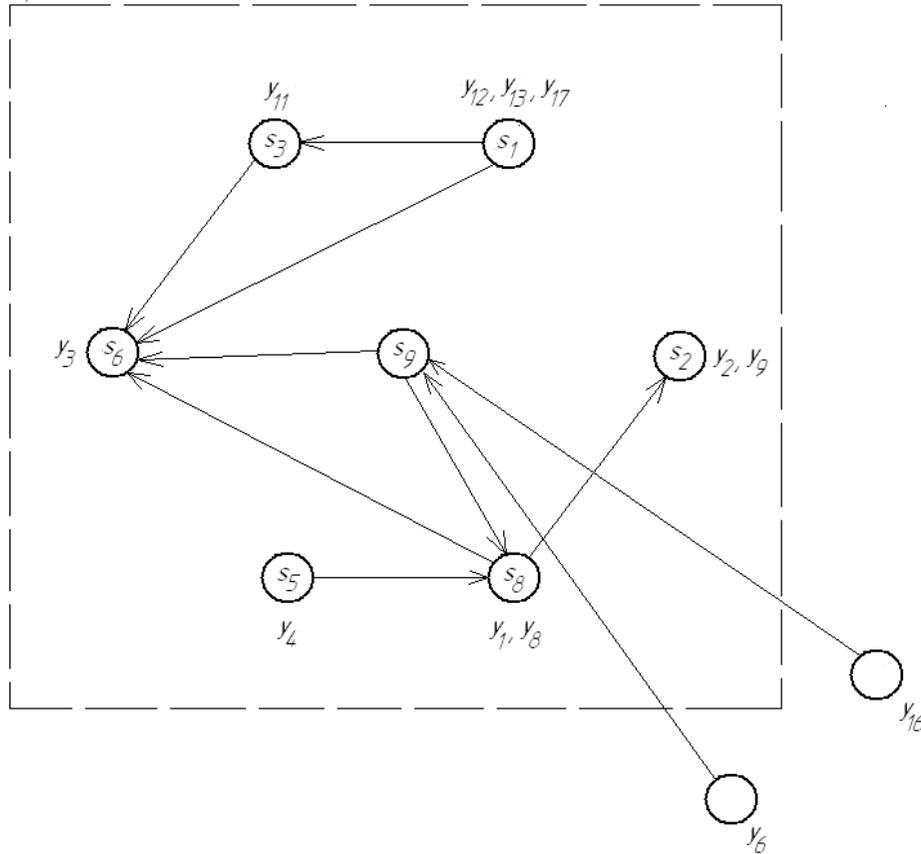


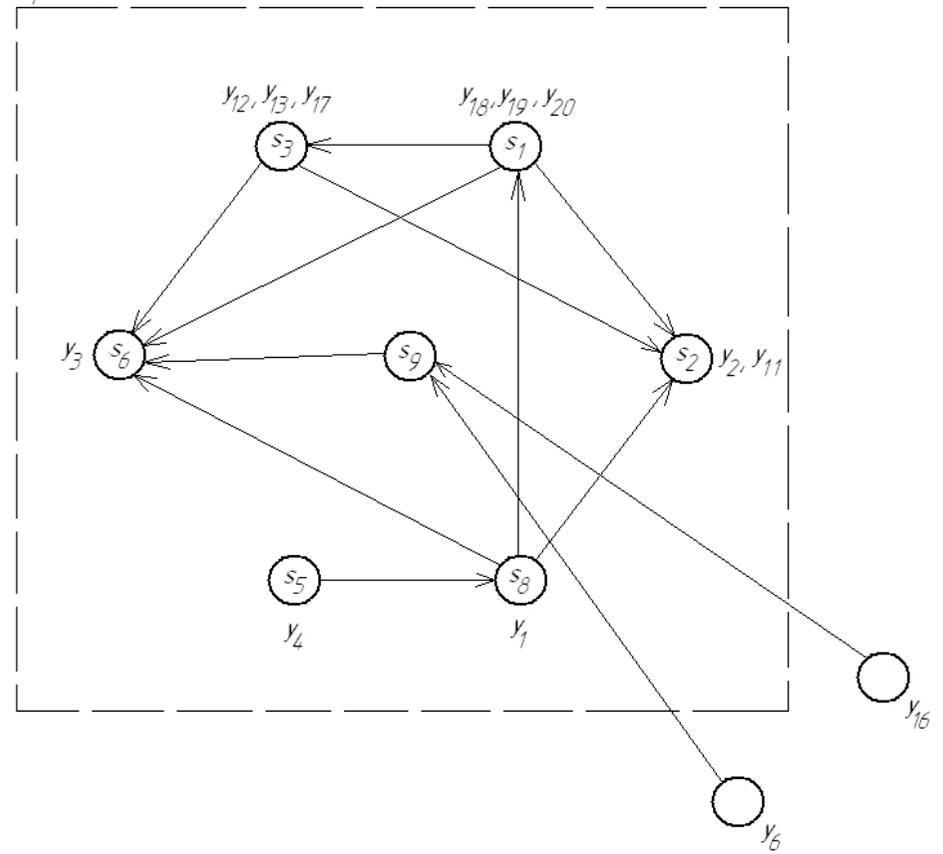
Рисунок 3.7 – Блок-схема алгоритма смешанной реконфигурации КБО ИМА

Крейт ИМА



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

Крейт ИМА



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

Рисунок 3.8 – Результат работы алгоритма смешанной реконфигурации КБО

3.4 Выводы

1. Структура КБО ИМА и три базовых категории возможных отказов (неправильное функционирование приложения или аппаратного устройства; неправильная реакция на действие исправного приложения или модуля; несвоевременное выполнение какой-либо функции системой ИМА) определяют возможность и необходимость проведения следующих видов реконфигурации системы: аппаратной, программной, функциональной, смешанной.

2. Определен вектор текущего состояния системы ИМА, значение которого показывает необходимость проведения одного из видов реконфигурации. Значение вектора текущего состояния системы ИМА автоматически формируется при ее работе на основе информации от встроенного контроля о количестве неисправных модулей, функциональных приложений и требуемого режима работы. Предложен алгоритм принятия решения о проведении реконфигурации на основе значения вектора текущего состояния системы ИМА.

3. Разработаны алгоритмы динамического изменения конфигурации системы ИМА: аппаратной, программной и смешанной реконфигурации.

4. Определены условия проведения функциональной реконфигурации системы – специального целенаправленного изменения ее конфигурации с целью повышения качества выполнения определенной задачи, например, перехода в режим, когда одна информация (данные от обзорной системы, системы управления вооружением и т.д.) становится более приоритетной, чем другая (например, навигация, автопилот и т.д.). Отличительной особенностью функциональной реконфигурации является возможность проведения при отсутствии отказов в системе.

5. Предложен алгоритм смешанной реконфигурации системы ИМА. Данный алгоритм построен на объединении алгоритмов решения задачи оптимального распределения функций КБО и аппаратной реконфигурации при наличии нескольких режимов работы, и реализуется с целью оптимизации функционирования системы ИМА в сложившихся в полете условиях.

6. Для повышения эффективности выполнения реконфигурации системы необходимо заранее произвести расчет вариантов перераспределения функций для основных возможных складывающихся режимов полета и технического состояния системы и сохранить их в виде таблиц реконфигурации для последующего использования при переключении. Такой подход позволит избежать опасных ситуаций при изменении конфигурации и, в случае необходимости, обеспечить функционирование системы ИМА в аварийном режиме для совершения экстренной посадки.

7. Для сравнения уровней надежности и безопасности реконфигурирующегося КБО и комплекса традиционной федеративной архитектуры, а также для определения их соответствия требованиям ТЗ в авиации, необходимо провести анализ возможных неисправностей системы ИМА и определить ее показатели надежности.

4 ИССЛЕДОВАНИЕ РЕКОНФИГУРИРУЮЩЕГОСЯ КБО ИМА

4.1 Оценка результатов работы алгоритмов оптимального распределения и реконфигурации

Предложенный алгоритм оптимального распределения по критерию минимума загрузки сети передачи данных лежит в основе алгоритмов всех типов реконфигурации [86]. Для оценки результатов его работы был проведен анализ, заключающийся в определении влияния значений и количества входных данных на размерность проектируемой системы. В результате рассмотрения ТЗ на различные авиационные системы и их конечных функциональных структур установлено следующее:

1. Количество информационных связей, соответствующих потокам данных между функциями, приблизительно равно числу этих функций.
2. Объем принимаемых и передаваемых функциональным приложением данных прямо пропорционален его сложности.
3. Отдельные функциональные приложения занимают в среднем 40-50% вычислительной мощности процессорных модулей.
4. Самые сложные функциональные приложения занимают не более 80% вычислительной мощности процессорных модулей с целью обеспечения возможных доработок в будущем.

Данные особенности позволяют проверить работу алгоритма распределения на лишь ограниченном объеме входных данных. Перебор всех возможных вариантов количества и сложности функций, потоков данных и режимов работы является не только очень трудоемкой вычислительной задачей, но и не дает необходимой информации с точки зрения реализации алгоритма на практике.

Таким образом, исследование алгоритма распределения было произведено с учетом следующих ограничений на входные данные:

- матрица смежности $A(G_F)$ строится на основе не более чем 35 функций;

- матрица потоков данных $D(G_F)$ в каждой строке содержит не более двух потоков данных;
- все потоки данных $v_m\sigma_m$ возможно отсортировать по убыванию только одним способом (все потоки не равны между собой);
- используется показатель средней вычислительной сложности функциональных приложений, которая задается целым числом из набора $\{20, 40, 60, 80\}$ в процентах от соответствующей производительности ВМ. Полагается, что все приложения обладают средней вычислительной сложностью;
- условие достаточности объема памяти ВМ (2.9) выполняется всегда.

Данные ограничения являются некритичными с точки зрения проектирования реальной системы, но в то же время могут дать необходимую информацию в достаточном для принятия решения объеме.

На первом этапе исследования оценивалось количество необходимых аппаратных модулей $v \in V$ для реализации различных наборов функций $y \in Y$, распределенных по предложенному алгоритму, при фиксированном значении средней сложности. Результат представлен на рисунке 4.1.

Как видно из рис. 4.1, зависимость числа ВМ от количества приложений практически линейная. При этом множество кривых можно условно разделить на две части. При средней сложности более 50% происходит качественный «скачок», при котором требуемое количество модулей резко возрастает. Это объясняется тем, что значение сложности приложений более 50% не позволяет в большинстве случаев определить более одной функции в вычислитель из-за недостаточной мощности последнего. При этом видно, что при стремлении средней сложности к 100%, количество модулей стремится к количеству приложений.

На втором этапе исследования оценивалась зависимость количества модулей $v \in V$ от их средней сложности при фиксированных наборах функциональных приложений $y \in Y$. Результат представлен на рисунке 4.2.

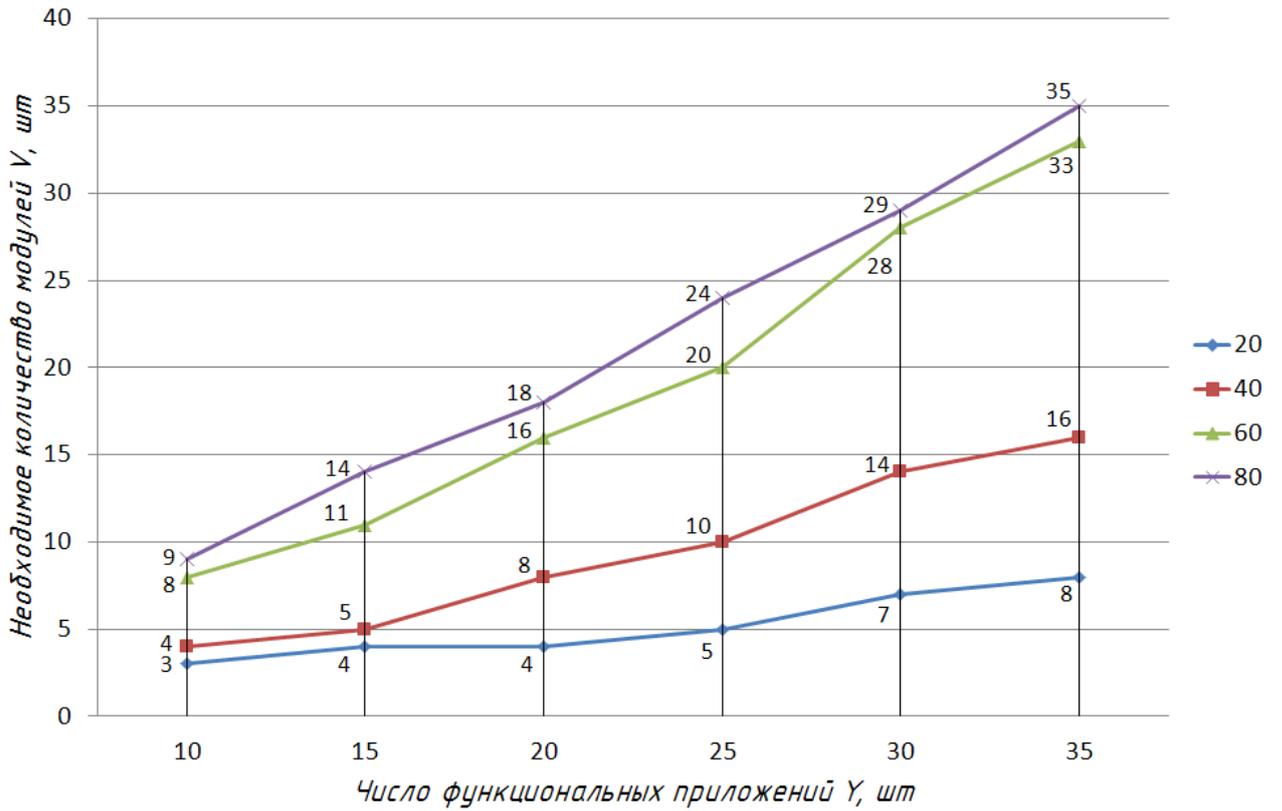


Рисунок 4.1 – Зависимость количества VM крейта от числа приложений при распределении по оптимальному алгоритму

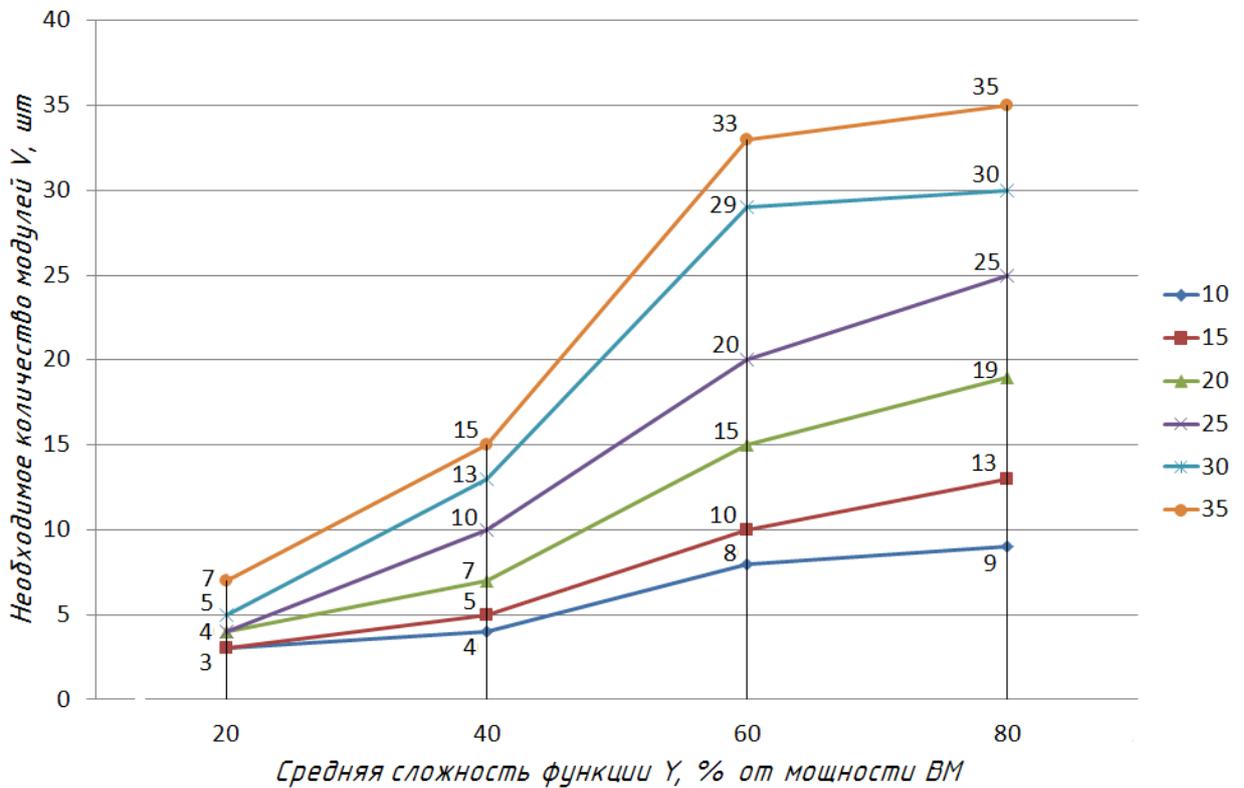


Рисунок 4.2 – Зависимость количества VM крейта от показателя средней сложности приложений при распределении по оптимальному алгоритму

Как видно из рис. 4.2, зависимость числа ВМ от показателя средней сложности также практически линейная. При этом наблюдается «скачок» в окрестности значения, близкого к 50%.

На основе кривых, изображенных на рис. 4.1 и 4.2 можно построить приближенную поверхность, отражающую зависимость между тремя рассматриваемыми параметрами (см. рисунок 4.3).

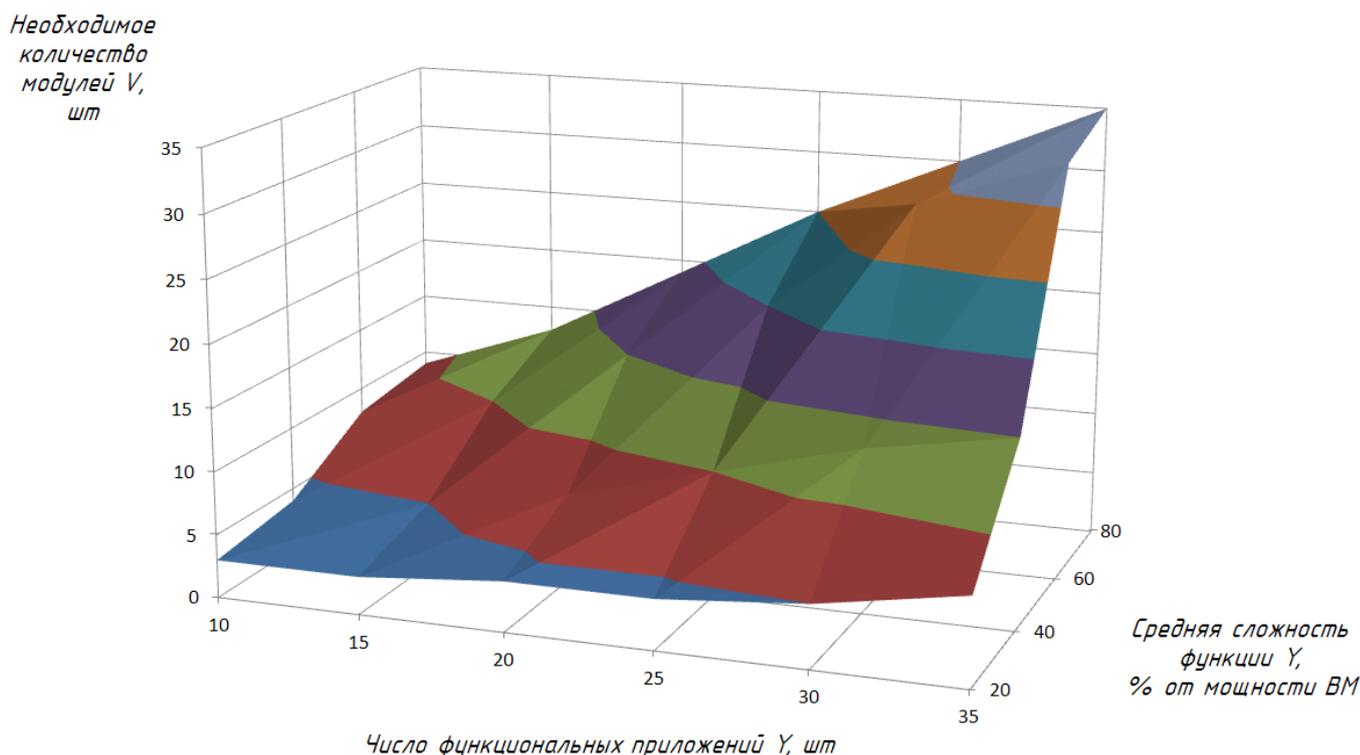


Рисунок 4.3 – Параметрическая поверхность КБО, полученная в результате работы алгоритма оптимального распределения

Основываясь на полученной параметрической поверхности можно оценить конечную размерность системы, зная количество функциональных приложений и их среднюю сложность, которая может быть определена посредством экспертных оценок. Так для КБО вертолета, функциональный граф которого изображен на рисунке 2.6, состоящего из 17 функций потребуется 4-5 модулей, что подтверждается графом аппаратной структуры, изображенным на рисунке 2.7, в котором для приложений были заданы конкретные значения сложностей исходя из анализа требований ТЗ. При этом количество ВМ в соответствующем графе равно 8 из-за трех функций, требующих всю вычислительную мощность модулей.

Оценка эффективности алгоритмов динамического управления конфигурацией проводилась на алгоритме аппаратной реконфигурации, поскольку остальные ее виды представляют собой соответствующие обобщения, не вносящие дополнительных параметров, требующих учета при перераспределении задач.

Алгоритм аппаратной реконфигурации учитывает критичность функций при их исключении из состава КБО. Поэтому на третьем этапе исследования оценивалась зависимость допустимого количества отказавших модулей $v \in S_k$ от количества функций с уровнями критичности k_1 и k_2 (уровни А и В с соответствии с КТ-178В). При этом данная зависимость рассматривалась для параметрической поверхности КБО, изображенной на рисунке 4.3. Результаты исследования приведены на рисунке 4.4.

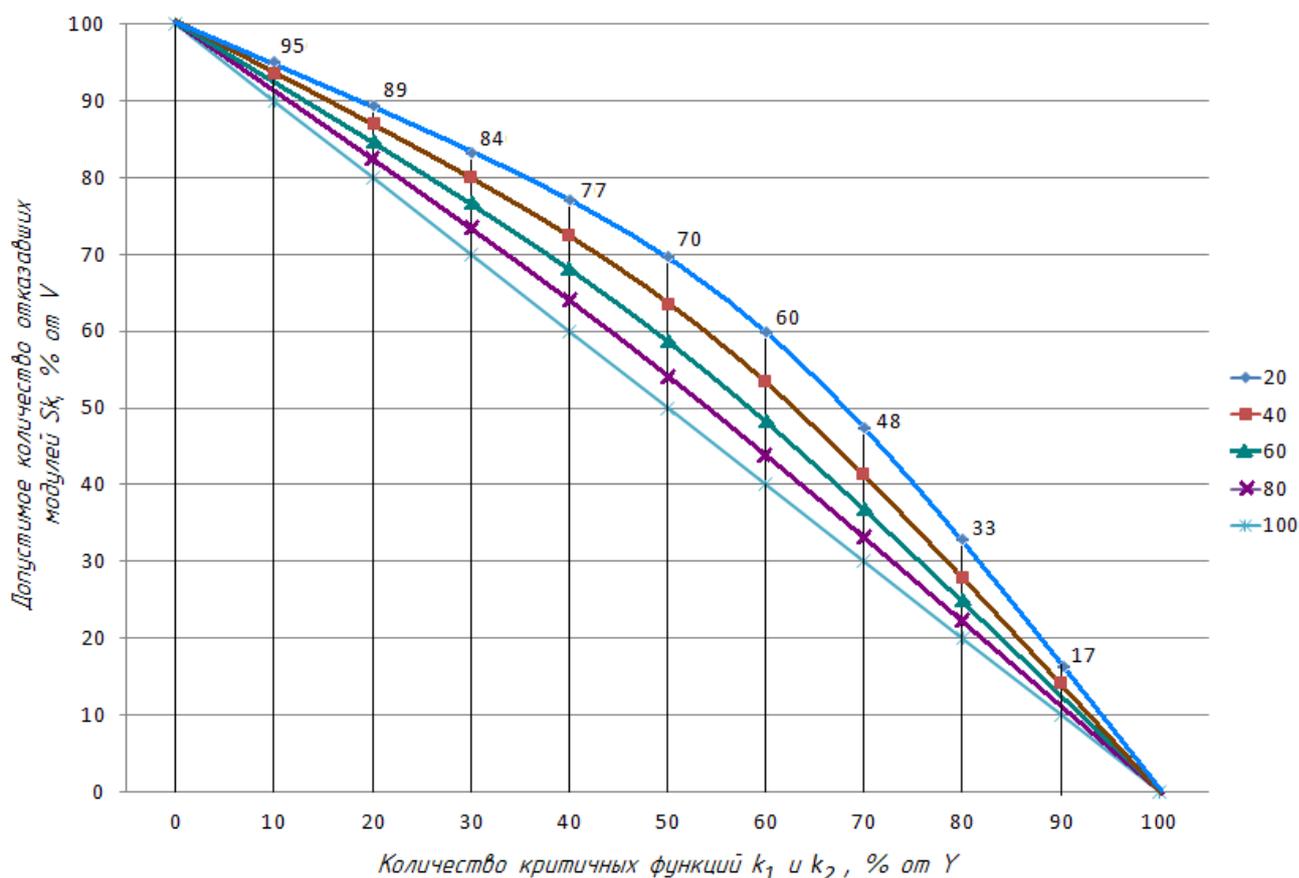


Рисунок 4.4 – Зависимость допустимого количества отказавших ВМ от количества функций с высокими уровнями критичности

Как видно из рис. 4.4 допустимое количество отказавших ВМ мало зависит от средней сложности функций в системе при распределении по предложенному ал-

горитму. Максимальная разница наблюдается при 50% функций с уровнями k_1 и k_2 , при этом допустим отказ дополнительно 20% модулей.

В результате исследования предложенных алгоритмов можно сделать следующие выводы:

- наибольшее влияние на размерность системы оказывает сложность функциональных приложений;

- влияние количества приложений на размерность системы усиливается с ростом их сложности;

- наибольшее влияние на допустимое количество отказавших ВМ оказывает доля приложений с высокими уровнями критичности. При этом влияние их сложности усиливается лишь на ограниченном участке (см. рис. 4.4).

- максимальная эффективность предложенных алгоритмов распределения и реконфигурации будет иметь место при большом количестве ВМ и функциональных приложений, не более 50% из которых обладают высокими уровнями критичности.

Все результаты исследований алгоритмов были получены на платформе .NET с использованием языка программирования высокого уровня С#.

4.2 Оценка безопасности и надежности реконфигурирующегося КБО ИМА

Оценка надежности КБО ИМА имеет ряд особенностей, которые связаны как с их структурой, так и с механизмами обеспечения реконфигурации [87]. Процесс оценки безопасности систем ИМА из-за высокой степени интеграции компонентов в соответствии с Р-297 должен проводиться по нормативам Р-4754 и Р-4761. При этом он должен учитывать следующие факторы [58]:

1. Изоляцию программных приложений друг от друга. Этот механизм обеспечивается самой архитектурой ИМА в соответствии с ARINC 653.

2. Защиту от множественных отказов (мониторинг рабочего состояния, управление неисправностями), которую обеспечивают предложенные механизмы и алгоритмы управления конфигурацией.

3. Предварительную оценку функциональной опасности, которая проводится в составе ЛА в целях проверки возможностей архитектуры ИМА для реализации заданных функций и назначения их уровней критичности.

4. Предварительную оценку безопасности (Preliminary System Safety Assessment – PSSA), которая необходима для определения перечня отказных состояний комплекса, а также условий их возникновения.

На первом этапе проведения PSSA проектируемого реконфигурирующегося КБО необходимо определить выполняемые им функции, состав и структуру аппаратной платформы, а также режимы ее работы [88, 89]. Платформа КБО представляет собой набор базовых ВМ, размещенных в едином крейте, и может быть описана неориентированным графом $G_S=(V, W)$ (см. рис. 2.7). При этом каждый модуль выполняет какой-либо набор функциональных приложений $y \in Y$.

Платформа может работать в двух основных режимах – рабочем и технологическом [88]. В рабочем режиме выполняются функции $y \in Y$, а также проводится текущий контроль собственной исправности. Рабочий режим подразделяется на несколько подрежимов, в которые комплекс переходит при осуществлении какого-либо вида реконфигурации. Количество подрежимов Q задается в ТЗ и заранее известно. Количество подрежимов, соответствующих каждому виду реконфигурации, задается в виде матрица реконфигурации. Перечень функций $y_q \in Y_q$, выполняющихся в подрежиме $q \in Q$, так же определен и представлен в виде неориентированного графа функциональной структуры $G_{Fq}=(Y_q, Z_q)$, где $z \in Z_q$ – дуга, соответствующая информационной связи между двумя функциями из множества Y_q . Обобщенная схема режимов работы платформы и условия перехода между ними представлены на рисунке 4.5.

Из режима «Выключен» после подачи питания КБО переходит в режим запуска, инициализирует программные и аппаратные компоненты, проводит встроенный контроль исправности и переходит по заданным условиям в рабочий или технологический режимы (загружается и запускается рабочий или технологический образы операционной системы реального времени – ОСРВ). В технологическом режиме платформа запускает системный раздел, обеспечивающий возмож-

ность загрузки данных в энергонезависимую память. При обнаружении в любом из режимов встроенным контролем неисправностей, которые могут повлиять на безопасное выполнение функциональных приложений, платформа запускает процесс реконфигурации в соответствии с выбранным критерием оптимальности и текущим состоянием комплекса, которое характеризуется дополнительными параметрами, такими как количество отказавших модулей $m \in S_k$ и приложений $y \in N_y$, а также установленным режимом полета $q \in Q$. Реконфигурация продолжается до тех пор, пока не будет исчерпан аппаратный ресурс, необходимый для переназначения задач, либо не произойдет такое количество отказов, при котором дальнейшее функционирование комплекса (выполнение задач с уровнями критичности k_1 и k_2 , $k \in K$) невозможно. В этом случае выставляется сигнал неисправности КБО.

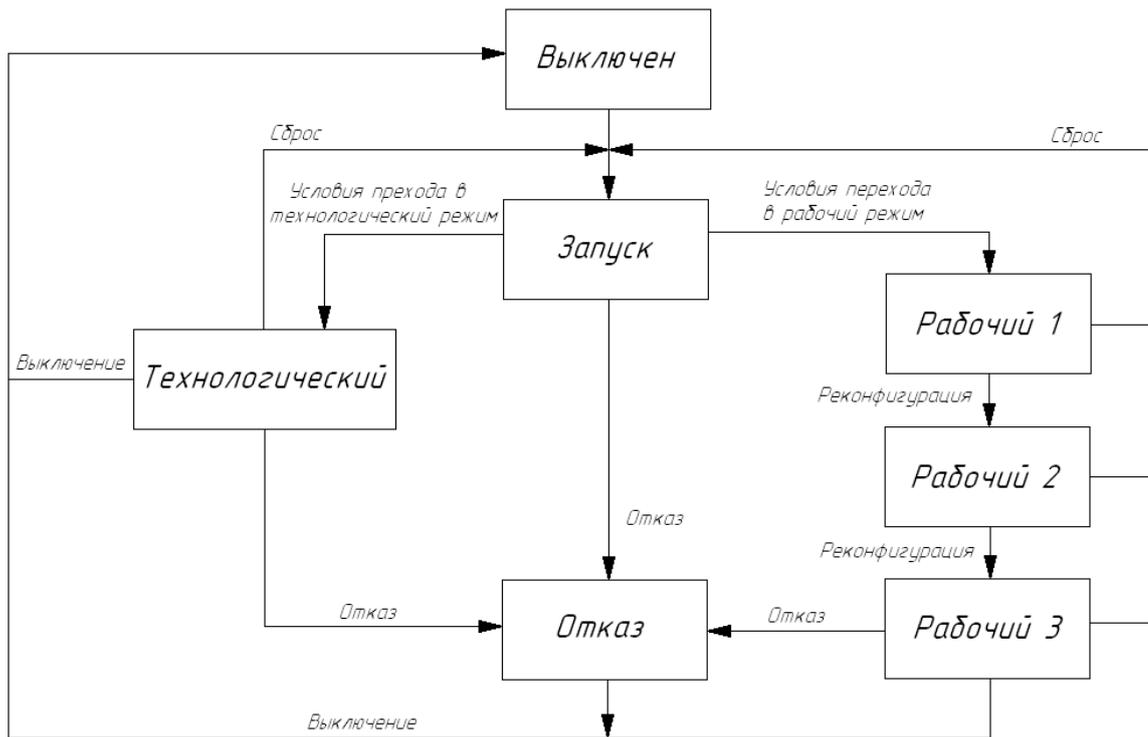


Рисунок 4.5 – Обобщенная схема режимов работы реконфигурирующегося КБО ИМА [88]

На втором этапе определяется критичность отказных состояний комплекса. Согласно АП-25 [90], для отказных состояний системы справедлива классификация, приведенная в таблице 4.1, и соответствующие ей численные показатели ве-

роятности возникновения. Такая классификация удовлетворяет требованиям стандартов на конструирование аппаратуры КТ-254 и ПО КТ-178В, в которых определены 5 уровней критичности выполняемых функций $k \in K$. При этом в КТ-254 не приведены конкретные числовые значения безотказности аппаратуры системы. Если аппаратура содержит в себе элементы, выполняющие функции с разными уровнями гарантии конструирования (критичности), стандарт предписывает использовать один из двух методов:

- вся аппаратная часть должна отвечать самому высокому уровню критичности;
- допускается принятие различных уровней для элементов при том условии, что критичные функции надежно защищены от влияния отказов менее критичных функций.

Таблица 4.1 – Классификация отказных состояний

Классификация отказного состояния по АП-25/CS-25	Обозначение	Вероятность возникновения	Численные показатели (на час полета), не менее
Без последствий/ No safety effect	БП	-	-
Усложнение условий полета/ Minor	УУП	Вероятные/ Probable	от $1 \times E-03$ до $1 \times E-05$
Сложная ситуация/Major	СС	Маловероятные/ Remote	от $1 \times E-05$ до $1 \times E-07$
Аварийная ситуация/ Hazardous	АС	Крайне маловероятные/ Extremely Remote	от $1 \times E-07$ до $1 \times E-09$
Катастрофическая ситуация/ Catastrophic	КС	Практически невероятные/ Extremely Improbable	менее $1 \times E-09$

Поскольку КБО ИМА, помимо базовых ВМ, состоит из специальных модулей, выполняющих функции с разными уровнями критичности и не участвующих в реконфигурации, то их повреждение повлечет за собой отказ всех выполняющихся на них приложений. Поэтому целесообразно выбрать первый метод и присвоить всему КБО максимальный уровень гарантии конструирования.

На третьем этапе производится построение и анализ деревьев неисправностей комплекса.

Анализ дерева неисправностей – дедуктивный анализ отказов, который сосредотачивается на одном конкретном нежелательном событии и позволяет определить причины этого события. Анализ дерева неисправностей – «нисходящая» процедура оценки системы, в которой формируется и оценивается качественная (дерево), а затем и количественная (значение вероятности отказа) модель для отказного события.

Каждое конечное событие дерева неисправностей непосредственно является полным отказом одного функционального узла, либо определенным видом частичного отказа внутри него. Вероятность отказа функционального узла определяется в соответствующем расчете надежности [91].

Для двух связанных перекрестными связями крейтов ИМА, структура которых изображена на рисунке 2.7, конечным (базовым) событием дерева будет отказ какого-либо ВМ. Отказ программного приложения не является базовым, поскольку при одиночных сбоях механизмы ОСРВ позволяют перезапустить отказавшее приложение. При этом отказ какого-либо элемента или датчика вне крейта, данные от которого являются входными для приложения, не относится непосредственного к крейту ИМА, а принадлежит КБО всего ЛА. Поскольку датчики в системах ИМА и федеративной архитектуры идентичны, то и их влияние на результирующую надежность одинаково. Это позволяет исключить их из рассмотрения.

Так как ПО авиационных комплексов разрабатывается по уровню А в соответствии с КТ-178В, вероятность необнаруженной ошибки в коде, приводящей к соответствующему событию в дереве неисправностей, считается событием практически невероятным (менее 1×10^{-9}). Поэтому данные виды отказов при анализе не включались в состав дерева ввиду их малого влияния на результирующую оценку.

Чтобы грубо оценить эффективность реконфигурирующихся крейтов, были построены и проанализированы деревья неисправностей для блоков классической федеративной архитектуры с двойным резервированием и архитектуры ИМА, состоящей из одного крейта с резервированным коммутатором.

Дерево неисправностей для федеративной архитектуры (для одного блока) изображено на рисунке 4.6, а для реконфигурирующегося крейта ИМА – на рисунке 4.7.

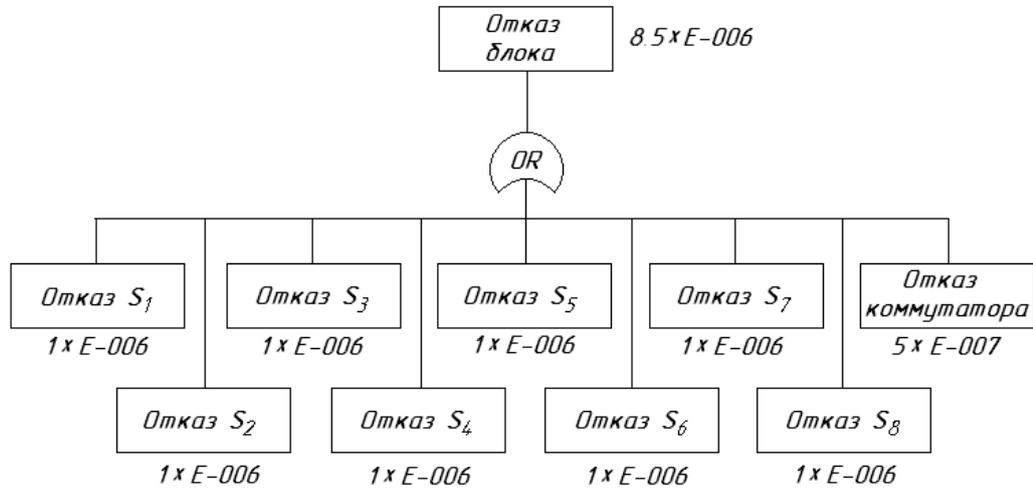


Рисунок 4.6 – Дерево неисправностей блока федеративной архитектуры [91]

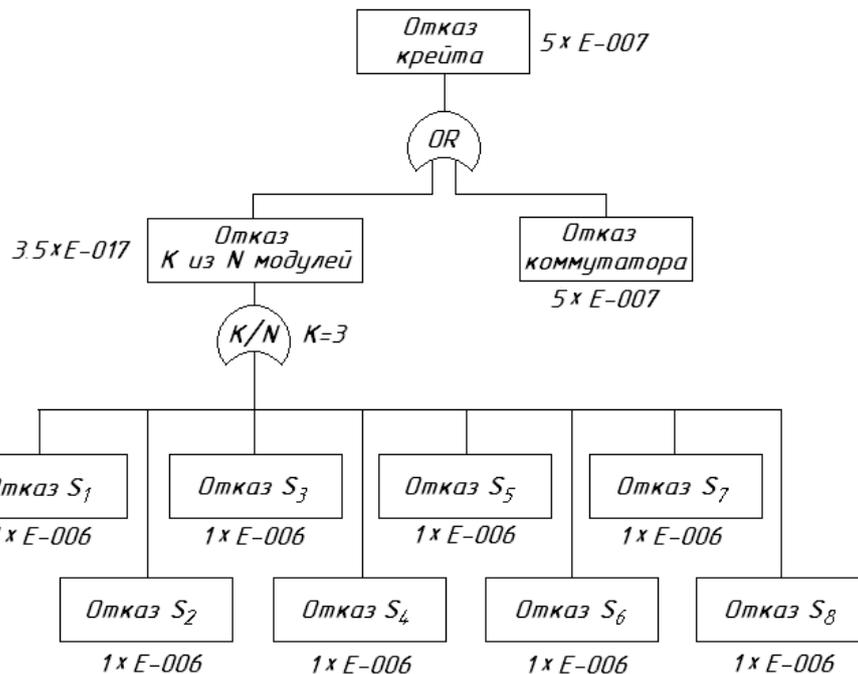


Рисунок 4.7 – Дерево неисправностей крейта архитектуры ИМА [91]

Поскольку все базовые ВМ идентичны друг другу, а отличия их от других типов модулей (например, модулей-концентраторов) в части интенсивности отказов минимальны (одного порядка), то интенсивность отказов каждого модуля λ_S принимается равной $1 \times E-006$. Так как коммутатор является самым уязвимым местом во всей системе, то он, как правило, резервируется дополнительно или проектиру-

ется из элементов повышенной надежности. Поэтому при расчетах его интенсивность отказов была принята $5 \times E-007$. Полный отказ федеративной системы (снятие сигнала исправности) происходит при отказе хотя бы одного модуля [91].

Отказ крейта ИМА, аппаратная структура которого представлена на рисунке 4.6 и который поддерживает возможность управления конфигурацией 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, по приблизительным оценкам, четыре ВМ (см. рис. 4.4).

Как видно из рисунков 4.6 и 4.7, 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). При этом стоит отметить, что коммутатор, фактически, является наиболее «узким» местом в крейте [91, 92].

Для более точной оценки надежности реконфигурирующегося КБО ИМА были построены и проанализированы деревья неисправности для распределенной системы ИМА, разработанной АО «УКБП» при участии автора и для варианта реализации этой системы на основе реконфигурирующегося крейта. Данная работа выполнена в рамках следующих СЧ ОКР:

- «Создание образцов комплекса унифицированных базовых элементов бортового оборудования открытой архитектуры на основе интегральной модульной авионики» по теме «Разработка курсоверткали»;

- «Разработка модуля воздушных данных МВД».

КБО ЛА, как правило, выполняет следующие функции, необходимые для успешного завершения полетного задания:

1. Задача навигационного обеспечения.
2. Формирование данных для отображения в системе индикации.

3. Регистрация массива полетной информации.
4. Прием и преобразование воздушных сигналов (давления, температуры торможения и т.д.).
5. Вычисление воздушных скоростей.
6. Управление кондиционированием и обогревом узлов ЛА.
7. Осуществление взаимодействия между различными системами ЛА.
8. Управление питанием (модули питания в соответствующих блоках и системах).
9. Контроль параметров и исправности силовой установки.
10. Формирование и выдача результатов контроля в системы индикации.

Таким образом, в состав распределенного КБО ИМА были включены следующие наборы блоков, выполняющих приведенные выше функции:

- блок преобразования сигналов БПС (рис. 4.8) – 3 шт.;
- блок вычислитель БВ, который осуществляет вычисление параметров в соответствии с типом данных, поступающих на его входы – 2 шт.;
- блок кондиционирования и обогрева БКО – 2 шт.;
- блок курсовертикали БКВ (рис. 4.9) – 1 шт.
- датчик магнитного курса ДМК (рис. 4.9) – 1 шт.;
- вычислитель воздушных сигналов ВВС – 3 шт.;
- модуль воздушных данных МВД – 6 шт.;
- приемник полного давления ППД – 4 шт.;



Рисунок 4.8 – Блок типа БПС производства АО «УКБП»

- приемник статического давления ПСД – 4 шт.;
- датчик аэродинамических углов ДАУ – 4 шт.;
- датчик температуры торможения ДТТ – 2 шт.

Функциональные узлы, выполняющие функции приема и преобразования воздушных сигналов приведены на рисунке 4.10.

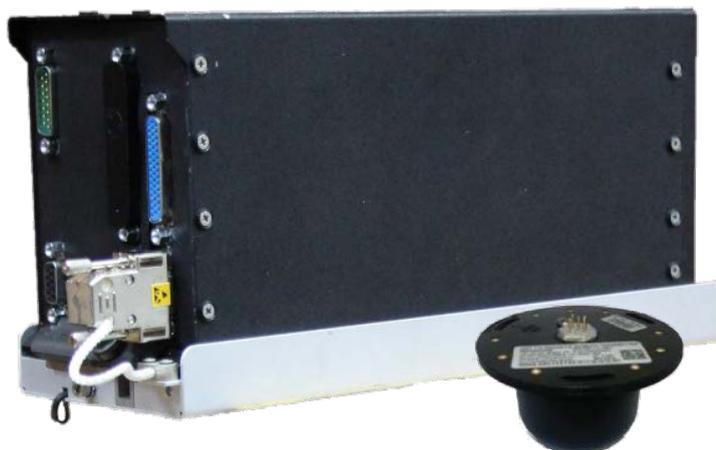


Рисунок 4.9 – Блок БКВ производства АО «УКБП» и датчик магнитного курса ДМК.

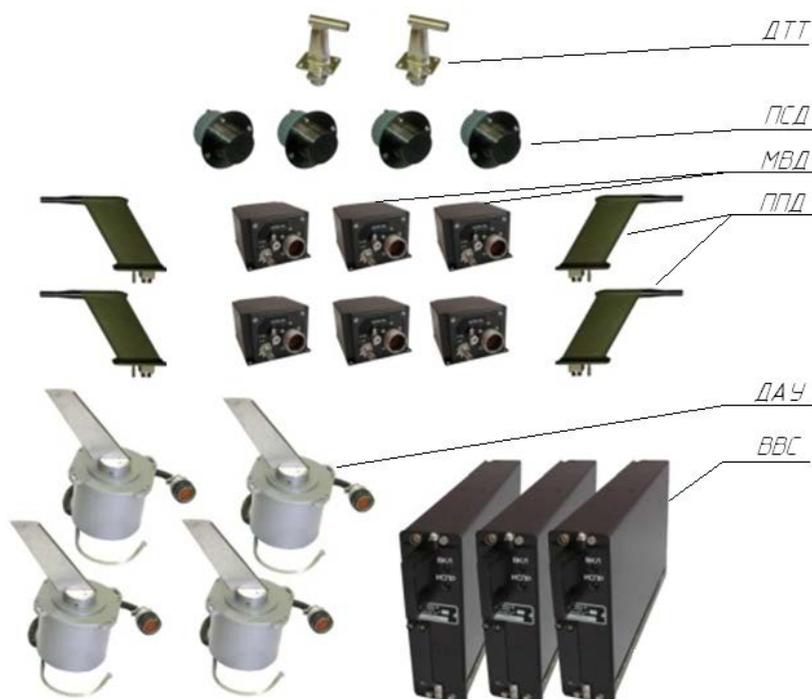


Рисунок 4.10 – Функциональные узлы, выполняющие функции приема и преобразования воздушных сигналов производства АО «УКБП»

На основании данных процесса управления конфигурацией, проведенного для аналогичных систем в соответствии с КТ-178В и КТ-254, критичность выполняемых функций определяется так, как представлено в таблице 4.2.

Таблица 4.2 – Показатели критичности для функций распределенной системы ИМА разработки АО «УКБП»

$k \in K$	Функции $y \in Y$ (сложность, % от мощности процессора ВМ)					
k_1	1 (40)	4 (50)	5 (30)	7 (60)	8 (30)	9 (30)
k_2	2 (60)					
k_3	3 (30)	6 (20)	10 (20)			

В результате анализа функционального программного обеспечения (ФПО) аналогичных систем, выполняющих указанные выше функции, были определены следующие входные данные для алгоритмов распределения и реконфигурации:

- матрица смежности $A(G_F)$:

$$A(G_F) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

При этом нулевые строки соответствуют функциям, данные от которых являются входными лишь для внешних систем или исполнительных элементов (индикации, регистратора и кондиционирования);

- матрица потоков данных $D(G_F)$:

$$D(G_F) = \begin{pmatrix} 0 & \nu_{1,2}\sigma_{1,2} & \nu_{1,3}\sigma_{1,3} & 0 & 0 & 0 & \nu_{1,7}\sigma_{1,7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \nu_{4,5}\sigma_{4,5} & \nu_{4,6}\sigma_{4,6} & 0 & 0 & 0 & 0 \\ 0 & \nu_{5,2}\sigma_{5,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \nu_{7,2}\sigma_{7,2} & \nu_{7,3}\sigma_{7,3} & 0 & 0 & 0 & 0 & 0 & 0 & \nu_{7,10}\sigma_{7,10} \\ 0 & 0 & 0 & 0 & 0 & 0 & \nu_{8,7}\sigma_{8,7} & 0 & \nu_{8,9}\sigma_{8,9} & 0 \\ 0 & \nu_{9,2}\sigma_{9,2} & 0 & 0 & 0 & 0 & \nu_{9,7}\sigma_{9,7} & 0 & 0 & 0 \\ 0 & \nu_{10,2}\sigma_{10,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- ограничение в виде неравенства 2.10 (по производительности процессора ВМ):

$$\beta_v \sum_{y \in Y} \nu_y \rho_y x_{yv} \leq 0,9 R_v, \forall v \in V.$$

Коэффициент 0,9 в данном неравенстве учитывает необходимые ресурсы процессора ВМ (10% вычислительной мощности) для работы приложений контроля.

- отношение потоков данных для распределения функций:

$$\begin{aligned} & \nu_{7,2}\sigma_{7,2} > \nu_{7,3}\sigma_{7,3} > \nu_{5,2}\sigma_{5,2} > \nu_{1,7}\sigma_{1,7} > \nu_{10,2}\sigma_{10,2} > \nu_{1,2}\sigma_{1,2} > \\ & > \nu_{9,7}\sigma_{9,7} > \nu_{9,2}\sigma_{9,2} > \nu_{8,9}\sigma_{8,9} > \nu_{8,7}\sigma_{8,7} > \nu_{1,3}\sigma_{1,3} > \nu_{7,10}\sigma_{7,10} > \nu_{4,6}\sigma_{4,6} > \nu_{4,5}\sigma_{4,5}. \end{aligned}$$

- один режим работы КБО, который предполагает проведение либо программной, либо аппаратной реконфигурации.

В результате работы предложенного алгоритма распределения по минимуму загрузки сети с дополнительным ограничением на загруженность ВМ в качестве выходных данных получена следующая матрица $R(G_S)$:

$$R(G_S) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Нулевой столбец 8 соответствует функции контроля и обеспечения питания, которая выполняется на специальном модуле в составе крейта, не участвующем в реконфигурации.

Необходимо отметить, что полученная матрица содержит лишь три пары функций, соединенные между собой информационной связью (строки 1, 3 и 4). Это объясняется малой размерностью проектируемого КБО и высоким процентом критичных функций при их одновременной сложности. В результате этого, требование по запасу вычислительной мощности ВМ выполнимо только при большем потоке данных по сети.

Средняя сложность функций (см. табл. 4.2) КБО составляет 38%. Количество функций с уровнями критичности k_1 и k_2 составляет 70%. Следовательно, на основании результатов исследования алгоритмов (см. рис. 4.4), допускается исключение 40% модулей, то есть двух модулей из первоначальных пяти, полученных в результате работы алгоритма распределения. Результат работы алгоритма аппаратной реконфигурации для двух отказавших модулей крейта представлен в виде матрицы:

$$R(G_S) = \begin{Bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{Bmatrix}$$

Нулевые столбцы соответствуют исключенным функциям 3, 6, 8 и 10. Таким образом, выполняется условие наличия запаса мощности ВМ и сохранения всех критичных приложений.

Для построения дерева неисправностей распределенной системы ИМА использовались схемы соединения соответствующих блоков (на основании документов с результатами проведения анализа видов и последствий отказов в аналогичных системах), а также информация по их интенсивностям отказов на основании расчетов надежности каждого вида блока, которые проводились по руководящему материалу РМ 25-2007 [93] в соответствии с ГОСТ 27.301 [94].

Анализ показал, что наибольшими интенсивностями отказов обладают элементы аналоговой части блоков. В частности, интенсивность отказов модуля

инерциальных датчиков МИД (см. рис. 4.11) из состава блока БКВ составляет $35,3118 \cdot 10^{-6}$ 1/ч, что на порядок больше, чем у более сложного модуля вычисления и формирования типа МВФ, изображенного на рисунке 4.12 ($3,7917 \cdot 10^{-6}$ 1/ч). Такой вклад в результирующее значение вносят три оптоволоконных датчика угловой скорости вращения.



Рисунок 4.11 – Модуль МИД разработки АО «УКБП»



Рисунок 4.12 – Модуль типа МВФ разработки АО «УКБП», применяющийся в качестве вычислителя в КБО ИМА

Резервирование в распределенной системе ИМА организовано следующим образом:

1. Трехкратным резервированием обладают блоки:

- БПС;
- ВВС.

2. Двукратным резервированием обладают блоки:

- БВ;

- МВД. Данные модули имеют возможность приема как статического, так и полного давления, и резервируются двукратно по каждому борту (левый и правый).

3. Не резервируется блок БКВ.

Значения интенсивностей отказов для каждого блока приведены в таблице 4.3.

Дерево неисправностей для распределенной системы ИМА представлено в приложении А.

Для построения дерева неисправностей реконфигурирующегося КБО ИМА схема соединения элементов формировалась с учетом следующих положений:

- поскольку во всех блоках распределенной системы в качестве вычислителя используются идентичные модули типа МВФ, то реконфигурирующийся крейт также будет состоять из пяти модулей МВФ с соответствующей интенсивностью отказов (см. табл. 4.3);

Таблица 4.3 – Интенсивности отказов блоков распределенной системы ИМА

Наименование блока/модуля	Интенсивность отказов, $\lambda \cdot 10^{-6}$ 1/ч
БКВ	53,6717
МП (модуль питания из состава БКВ)	10,7716
МВФ (из состава БКВ)	3,7917
МИД (из состава БКВ)	35,3118
БПС	7,2831
БВ	7,5824
МВД	39,2301
БКО	6,3657
ВВС	4,1356

- поскольку все вычислительные операции выполняются в крейте ИМА, то из состава МВД, БКВ и БВ исключаются модули типа МВФ, что приводит к соответствующему снижению интенсивности отказов блоков;

- в схему соединения вводятся два резервированных модуля питания, которые обеспечивают выдачу необходимого напряжения для всех приемников. Резерви-

рование необходимо для обеспечения заданного уровня критичности соответствующей функции;

- датчики статического и полного давления резервируются четырехкратно, что соответствует четырем модулям МВД. Датчик температуры торможения резервируется двукратно;

- модуль МИД резервируется двукратно из-за высокого показателя интенсивности отказов для обеспечения заданного уровня критичности соответствующей функции (навигационное обеспечение).

Дерево неисправностей для распределенной системы ИМА представлено в приложении Б.

На основании деревьев неисправностей был проведен расчет вероятности отказа распределенной системы ИМА и реконфигурирующегося крейта ИМА на один час полета для экспоненциального закона распределения вероятностей [95]:

$$P(t) = 1 - Q(t) = e^{-\lambda t}.$$

Расчет проводился в системе RAM Commander v. 8.4 с параметрами относительной величины отсечения $1 \times E-006$, точности оптимизации 0,001 и продолжительности полета 1 ч. Вероятность отказа реконфигурирующегося крейта составила $2,78 \cdot 10^{-9}$, а распределенной системы ИМА $5 \cdot 10^{-5}$.

Анализ промежуточных деревьев показал, что наибольший вклад в результирующую вероятность отказа вносят нерезервированные элементы с высокой λ (БКВ). При этом двойное резервирование помогает достичь требований по безопасности к системе уровня А по КТ-178В и КТ-254. Сосредоточение вычислительных мощностей в высоконадежном реконфигурирующемся крейте позволяет уменьшить показатель суммарной интенсивности отказов резервированных элементов (датчиков), что в свою очередь приводит к уменьшению вероятности отказа КБО.

При этом стоит отметить, что в случае контролируемого отказа распределенной системы ИМА в комплексе существует возможность переключения на интегрированную систему резервных приборов (ИСРП), которая фактически сосредото-

тачивает в себе наиболее критичные функции и «резервирует» остальную систему. Это позволяет выполнить требования стандартов по уровню А.

С точки зрения стоимости конечного КБО реконфигурирующийся крейт позволяет использовать менее надежные элементы при проектировании ВМ за счет проводимой реконфигурации, так как его вероятность отказа вносит очень малый вклад в результирующий показатель надежности (на 1-2 порядка меньше, чем датчики и исполнительные элементы). Более того, крейт позволяет значительно сократить количество типовых ВМ в системе. Так для рассматриваемого КБО число модулей типа МВФ сократилось почти на 50%, что позволяет дополнительно зарезервировать низконадежные датчики, такие как МИД, при одновременном снижении стоимости системы.

4.3 Методика предварительного определения конфигурации КБО ИМА

Процесс разработки аппаратной платформы ИМА является частью общего процесса проектирования системы ИМА в соответствии с Р-297, наряду с программными приложениями и их интеграцией. При этом подчеркивается, что платформа может быть разработана отдельно от конкретных функций ЛА и повторно использоваться с другим набором поддерживаемых приложений. Однако при создании КБО первичным является ТЗ, содержащее в себе перечень выполняемых функций и лишь общие требования к аппаратному обеспечению (протоколы взаимодействия с другими системами, габариты, масса и т.д.). Поэтому обобщенная методика предварительного определения аппаратной конфигурации КБО ИМА должна основываться на системном ТЗ и включать в себя следующие шаги:

1. Первичная декомпозиция функций, заданных в ТЗ, на множество программных приложений $y \in Y$ и функций, выполняемых аппаратной частью, которая осуществляется на основе экспертных оценок и опыта проектирования ПО авиационных систем. При этом необходимо определить информационные связи между приложениями и их характеристики в соответствии с требованиями к ПО, которые, в общем случае, оформляются на основе описания аппаратных средств. Поскольку платформа еще не разработана, то требования к ПО должны учитывать ТЗ на аппаратную часть. Декомпозиция включает себя определение режимов работы системы и соответствующие им множества приложений, а также определяется и критичность выполняемых функций, которая, как правило, задана в ТЗ.

Эта процедура выполняется и в случае использования уже готовой платформы с той лишь разницей, что требования к ПО должны учитывать ее архитектуру.

Фактически, на этом шаге необходимо определить функциональный состав проектируемого КБО $G_F = (Y(K, P), Z(\Xi, \mathcal{G}))$ и соответствующие ему параметры.

2. Проведение исходного анализа деревьев неисправностей (ФТА) в контексте процесса определения функциональной опасности (ФНА) на уровне системы. Данный анализ необходим для распределения бюджета вероятностей по элементам системы.

3. Определение архитектуры платформы ИМА $G_S = (V(L, R), W(\Psi, \Omega))$, которое включает в себя тип и основные функции различных модулей, ресурсы и компоненты, порядок их взаимодействия (распределенная или централизованная архитектура), механизмы контроля собственной исправности (в том числе и программные в виде приложений контроля $y \in Y_{rec}$), а также характеристики сети передачи данных.

При наличии целевой платформы и дополнительных требований проверяется возможность их реализации путем задействования резервов (например, свободные входы мультиплексоров для организации большего числа каналов, свободные ресурсы ПЛИС и т.д.) или проведения доработок. Также проверяется соответствие уровня гарантии конструирования аппаратуры требуемому уровню критичности приложений.

4. Распределение функциональных задач по аппаратной платформе на основании проведенных оценок параметров проектируемого КБО. В качестве целевой функции и критерия оптимальности могут быть выбраны те, которые в наибольшей степени удовлетворяют требованиям ТЗ. Рекомендуется использовать критерий минимума загрузки сети и соответствующий ему алгоритм, приведенные в главе 2, так как они учитывают специфику ИМА и позволяют получить систему с приемлемой размерностью за ограниченное время.

5. Повторное проведение ФТА с модифицированными деревьями неисправности, которые получены с учетом выбранной архитектуры аппаратной платформы и заданного распределения функций. Полученная вероятность отказа сравнивается с соответствующим требованием безопасности и в случае его удовлетворения происходит фиксация проекта системы [61].

При отсутствии требований ТЗ по повышенному уровню надежности КБО, оценка его «базовой» структуры на этом завершается, а в случае наличия целевой платформы принимается решение о ее соответствии выставленным требованиям.

При дальнейшем изменении аппаратной или функциональной части или по результатам испытания опытного образца производится уточнение деревьев и повторное проведение ФТА.

6. Реализация реконфигурирующегося КБО. На этом этапе необходимо определить следующие параметры:

- перечень отказов, после которых необходимо проведение реконфигурации;
- перечень видов проводимой реконфигурации;
- таблицы реконфигурации для требуемых типов отказов, определенные посредством применения алгоритмов динамической реконфигурации.

При этом могут оцениваться лишь те параметры, которые в наибольшей степени влияют на функционирование и состав системы (например, функциональная и аппаратная реконфигурации). При проработке ситуаций типа «наихудший случай» можно получить довольно точное представление об изменениях состояния проектируемого КБО и, в рамках процессов по управлению конфигурацией ИМА, оценить влияние отказа любого элемента, входящего в его состав. После этого принимается решение о необходимости его парирования.

7. Модификация деревьев неисправностей с учетом механизмов управления конфигурацией, введение дополнительных структурных элементов в КБО (например, дополнительного резервирования нереконфигурируемых частей) и повторное проведение ФТА.

8. На основании полученных результатов на каждом из этапов проектирования производится выпуск или корректировка по сообщениям о проблемах следующих материалов процесса оценки безопасности в соответствии с планом управления конфигурацией системы ИМА [96]:

- анализ видов и последствий отказов ФТА;
- предварительная оценка безопасности PSSA;
- сводка видов и последствий отказов FMES;
- отчет по оценке безопасности SSA.

Схема процесса определения конфигурации КБО ИМА приведена на рисунке 4.13.

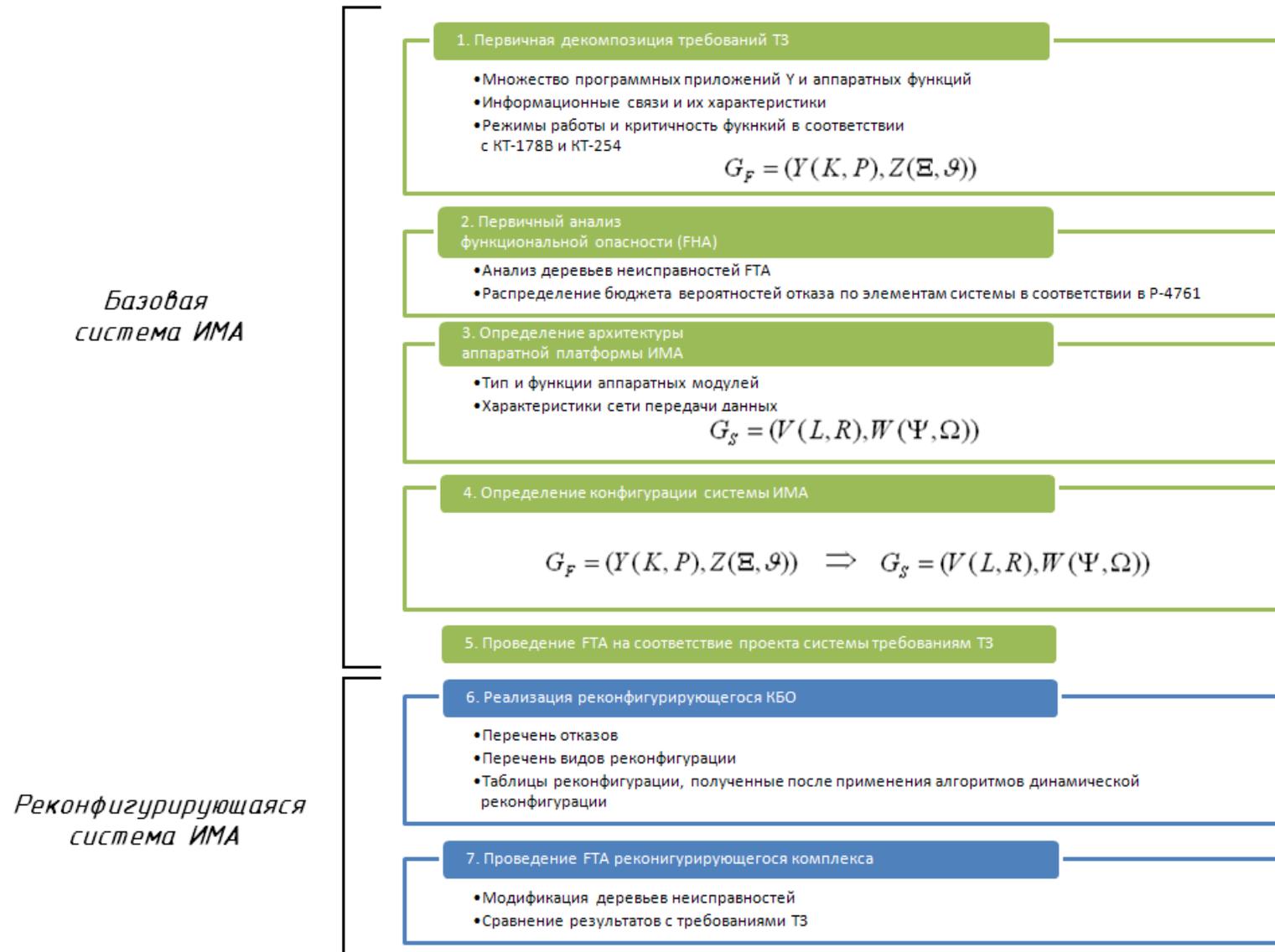


Рисунок 4.13 – Схема процесса определения конфигурации КБО ИМА

4.4 Выводы

1. Размерность проектируемого реконфигурирующегося КБО ИМА в большей степени зависит от сложности и количества выполняемых функций с уровнями критичности А и В в соответствии с «Руководством по гарантии конструирования бортовой электронной аппаратуры КТ-254» (RTCA DO-254/EUROCAE ED-80 "Design Assurance Guidance for Airborne Electronic Hardware") и Квалификационными требованиями КТ-178В «Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники» (RTCA/DO-178B «Software considerations in airborne systems and equipment certification»).

2. Алгоритм распределения программных приложений по минимуму загрузки сети передачи данных и соответствующая ему параметрическая поверхность позволяют с достаточной точностью оценить модульный состав проектируемого реконфигурирующегося КБО ИМА.

3. Исследования предложенных алгоритмов определения конфигурации ИМА показали, что для обеспечения возможности гибкого распределения функций по ВМ и для обеспечения возможности проведения реконфигурации системы критичным является соотношение сложности реализуемых функций и производительности ВМ. При этом:

- наибольшее влияние на размерность системы оказывает сложность функциональных приложений;
- влияние количества приложений на размерность системы усиливается с ростом их сложности;
- наибольшее влияние на допустимое количество отказавших ВМ оказывает доля приложений с высокими уровнями критичности;
- эффективность предложенных алгоритмов и гибкость системы ИМА возрастает при достаточном количестве функциональных приложений, сложность которых не превышает 10-30% от производительности ВМ;

- эффективность предложенных алгоритмов распределения и реконфигурации возрастает при достаточном количестве ВМ и функциональных приложений, не более 50% из которых обладают высокими уровнями критичности.

4. Максимальная эффективность предложенных алгоритмов распределения и реконфигурации будет иметь место при большом количестве ВМ и функциональных приложений. В этом случае крейт будет иметь достаточное количество потенциальных аппаратных ресурсов для распределения критичных функций в случае возникновения отказов.

5. Анализ деревьев неисправностей в соответствии с «Руководством P-4754 по сертификации сложных бортовых систем воздушных судов гражданской авиации» (SAE ARP-4754) и «Руководством P-4761 по методам оценки безопасности систем и бортового оборудования воздушных судов гражданской авиации» (SAE ARP-4761) показал, что вероятность отказа реконфигурируемого крейта ИМА более чем на порядок меньше, чем соответствующая вероятность отказа блока федеративной архитектуры.

6. Вероятность отказа за 1 час полета КБО ИМА, построенного на основе реконфигурируемого крейта ИМА, на несколько порядков ниже, чем у распределенной системы ИМА, состоящей из того же набора аппаратных средств и выполняющей тот же набор функциональных приложений.

7. Предложена обобщенная методика предварительного определения аппаратной конфигурации КБО ИМА, которая включает первичную декомпозицию функций системы, проведение исходного анализа деревьев неисправностей (Fault Tree Analysis – FTA) в контексте процесса определения функциональной опасности (Functional Hazard Assessment – FHA) на уровне системы, определение архитектуры платформы ИМА, распределение функциональных задач по аппаратной платформе на основании проведенных оценок параметров проектируемого КБО, повторное проведение FTA с модифицированными деревьями неисправности, которые получены с учетом выбранной архитектуры аппаратной платформы и заданного распределения функций, реализацию реконфигурирующегося КБО, модификацию деревьев неисправностей с учетом механизмов реконфигурации в

КБО, результирующую оценку его уровней надежности и безопасности. Показано использование разработанных в данной работе математических моделей и алгоритмов при реализации процесса проектирования КБО ИМА.

ЗАКЛЮЧЕНИЕ

Результаты, полученные при проведении исследований, позволяют сделать вывод, что цель диссертационной работы, а именно, развитие методов проектирования систем ИМА в соответствии с основными положениями концепции, разработка моделей и алгоритмов оптимального распределения ресурсов КБО и алгоритмов динамического управления структурой системы при возникновении отказов с целью обеспечения требуемых показателей надежности проектируемых систем, в основном достигнута.

При этом были получены следующие результаты:

1. Проведен обзор и анализ архитектуры, конструктивных особенностей и структурных элементов известных КБО ИМА, которые показали, что, несмотря на существующие в настоящее время различные подходы к реализации основных положений проектирования интегрированных модульных комплексов, все они обладают общим набором характерных свойств. При этом показано, что в настоящее время существует множество методов проектирования аппаратной части интегрированных комплексов, в которых, однако, проблемы оптимизации распределения функциональных задач либо не рассматривают вовсе, либо рассматриваются с позиции независимой разработки программной части.

2. Отмечено, что структура перспективного отказоустойчивого КБО ИМА определяется тремя взаимосвязанными компонентами: аппаратной платформой, функциональными приложениями и ОСРВ – которые являются взаимодополняющими и должны проектироваться с учетом особенностей интеграции и совместной реализации. При этом показано, что для построения аппаратной платформы комплексов ИМА широко используются COTS-компоненты, которые обеспечивают высокопроизводительную элементную базу при достаточной экономической эффективности. Для того чтобы КБО ИМА, спроектированный с использованием подобных компонентов, соответствовал авиационным нормам и успешно прошел испытания, необходимо при его разработке использовать специальные технологические и конструкторские методы. Отмечено, что одним из наиболее перспектив-

ных конструктивных методов повышения надежности проектируемого КБО ИМА является совместное решение задач оптимального использования ресурсов вычислительных модулей, распределения функциональных задач и управления конфигурацией комплекса при возникновении отказов или изменении решаемых задач, для чего необходимо разработать специальную математическую модель комплекса, которая объединит в себе все его структурные составляющие, позволит получать, анализировать и оценивать различные архитектурные и структурные варианты его построения.

3. Показано, что при проектировании КБО ИМА должны быть учтены такие характеристики сложной системы как иерархическая структура, наличие связей между структурными элементами, глубокая интеграция структурных элементов и необходимость их проектирования с учетом взаимного влияния друг на друга, а также возможность функционирования в различных режимах и наличие множества параметров, характеризующих как отдельные элементы, так и комплекс в целом. При этом в качестве критерия оптимальности при выборе конфигурации системы ИМА предложено использовать критерий минимума загрузки сети передачи данных, так как это позволит повысить эффективность выполнения функций и обеспечить полное или частичное их выполнение случае отказа сетевого коммутатора.

4. Разработана единая математическая модель, объединяющая аппаратную и программную составляющие системы ИМА. Модель отличается тем, что кроме связей между аппаратной и программной частями системы в ней учитываются особенности архитектуры аппаратной и организации программной частей. Модель позволяет разрабатывать алгоритмы управления конфигурацией систем ИМА с обеспечением требуемого уровня надежности системы в целом. Показано, что возможность определения базовой конфигурации на основе заданного критерия оптимальности с учетом критичности функциональных приложений и требуемого уровня гарантии конструирования аппаратуры с использованием разработанной модели системы ИМА на основе решения одной из типовых задач на графах (поиск максимального потока, задача о назначениях, задача об упаковке и

т.п.) сильно ограничена в силу архитектурных и структурных особенностей системы ИМА и приводит либо к потере существенных данных о конструкции системы, либо к значительным вычислительным затратам, не гарантирующим нахождение приемлемого решения.

5. Разработан алгоритм определения базовой конфигурации системы ИМА на основе заданного критерия оптимальности с учетом критичности функциональных приложений и требуемого уровня гарантии конструирования аппаратуры. Предложенный алгоритм учитывает большое число ограничений, что не приводит к возрастанию числа вариантов перебора и времени поиска решения, и может быть применен на самых ранних стадиях проектирования системы, когда известны лишь набор функций системы, требования к обмену информацией и характеристики ВМ. Применение разработанной модели и алгоритма определения базовой конфигурации системы ИМА рассмотрено на примере КБО вертолета, выполняющего 17 типовых полетных функций.

6. Рассмотрены три базовых категории возможных отказов КБО ИМА, которые определяют необходимость проведения аппаратной, программной, функциональной и смешанной реконфигураций. Определен вектор текущего состояния системы ИМА, значение которого показывает необходимость проведения одного из видов реконфигурации и автоматически формируется при работе системы на основе информации от встроенного контроля о количестве неисправных модулей, функциональных приложений и требуемого режима работы. Предложен алгоритм принятия решения о проведении реконфигурации в зависимости от значения вектора текущего состояния системы ИМА, а также алгоритмы аппаратной, программной и смешанной реконфигурации. Также определены условия проведения функциональной реконфигурация системы – специального целенаправленного изменения ее конфигурации для повышения качества выполнения определенной задачи, например, перехода в режим, 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.). Отмечено, что отличительной особенностью функциональной реконфигурации является воз-

возможность проведения при отсутствии в системе отказов. Показано, что для повышения эффективности выполнения реконфигурации системы заранее произвести расчет вариантов перераспределения функций для основных возможных складывающихся режимов полета и технического состояния системы и сохранить их в виде таблиц реконфигурации для последующего использования при переключении. Такой подход позволит избежать опасных ситуаций при изменении конфигурации и, в случае необходимости, обеспечить функционирование системы ИМА в аварийном режиме для совершения экстренной посадки.

7. Исследования предложенного алгоритма распределения показали, что размерность проектируемого реконфигурирующегося КБО ИМА в большей степени зависит от сложности и количества выполняемых функций с уровнями критичности А и В в соответствии с «Руководством по гарантии конструирования бортовой электронной аппаратуры КТ-254» (RTCA DO-254/EUROCAE ED-80 "Design Assurance Guidance for Airborne Electronic Hardware") и Квалификационными требованиями КТ-178В «Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники» (RTCA/DO-178В «Software considerations in airborne systems and equipment certification»). На основе алгоритма была получена параметрическая поверхность КБО, которая позволяет с достаточной точностью оценить модульный состав проектируемого реконфигурирующегося КБО ИМА. Показано, что для гибкого распределения функций по ВМ и для обеспечения возможности проведения реконфигурации системы критичным является соотношение сложности реализуемых функций и производительности ВМ. При этом:

- наибольшее влияние на размерность системы оказывает сложность функциональных приложений;
- влияние количества приложений на размерность системы усиливается с ростом их сложности;
- наибольшее влияние на допустимое количество отказавших ВМ оказывает доля приложений с высокими уровнями критичности;

- эффективность предложенных алгоритмов и гибкость системы ИМА возрастает при достаточном количестве функциональных приложений, сложность которых не превышает 10-30% от производительности ВМ;

- эффективность предложенных алгоритмов распределения и реконфигурации возрастает при достаточном количестве ВМ и функциональных приложений, не более 50% из которых обладают высокими уровнями критичности.

Исследования показали, что максимальная эффективность предложенных алгоритмов распределения и реконфигурации будет иметь место при большом количестве ВМ и функциональных приложений. В этом случае крейт будет иметь достаточное количество потенциальных аппаратных ресурсов для распределения критичных функций в случае возникновения отказов.

8. Построение и анализ деревьев неисправностей в соответствии с «Руководством R-4754 по сертификации сложных бортовых систем воздушных судов гражданской авиации» (SAE ARP-4754) и «Руководством R-4761 по методам оценки безопасности систем и бортового оборудования воздушных судов гражданской авиации» (SAE ARP-4761) показал, что вероятность отказа реконфигурируемого крейта ИМА более чем на порядок меньше, чем соответствующая вероятность отказа блока федеративной архитектуры. При этом вероятность отказа за 1 час полета реконфигурируемого КБО ИМА на несколько порядков ниже, чем у распределенной системы ИМА, состоящей из того же набора аппаратных средств и выполняющей тот же набор функциональных приложений. На основе полученных результатов исследования алгоритмов предложена обобщенная методика предварительного определения аппаратной конфигурации КБО ИМА, которая включает первичную декомпозицию функций системы, проведение исходного анализа деревьев неисправностей (Fault tree analysis - FTA) в контексте процесса определения функциональной опасности (Functional Hazard Assessment - ФНА) на уровне системы, определение архитектуры платформы ИМА, распределение функциональных задач по аппаратной платформе на основании проведенных оценок параметров проектируемого КБО, повторное проведение FTA с модифицированными деревьями неисправности, которые получены с учетом выбранной архи-

тектуры аппаратной платформы и заданного распределения функций, реализацию реконфигурирующегося КБО, модификацию деревьев неисправностей с учетом механизмов реконфигурации в КБО, результирующую оценку его уровней надежности и безопасности. Показано использование разработанных в работе математических моделей и алгоритмов при реализации процесса проектирования КБО ИМА.

Таким образом, положения, приведенные в диссертации, теоретически обоснованы и могут быть применены при проектировании перспективных реконфигурирующихся систем ИМА. Практическое использование результатов работы подтверждено актом об их внедрении в процесс разработки систем АО «УКБП», приведенным в приложении В.

СПИСОК ЛИТЕРАТУРЫ

1. Alena R., Ossenfort J., Laws K., Goforth A., Filed M., Figueroa F. Communications for Integrated Modular Avionics [Электронный ресурс]. – Режим доступа: <http://ti.arc.nasa.gov/m/pub-archive/1277h/1277%20%28Alena%29.pdf>, свободный. Яз. англ. (дата обращения 16.12.2015).
2. Grabowski G., Balser B., Forster M. Integrated Modular Avionics with COTS directed to Open System and Obsolescence Management [Электронный ресурс]. – Режим доступа: www.dtic.mil/dtic/tr/fulltext/u2/p010973.pdf, свободный. Яз. англ. (дата обращения 16.12.2015).
3. Black R. Using Proven Aircraft Avionics Principles to Support a Responsive Space Infrastructure [Электронный ресурс]. – Режим доступа: http://www.responsivespace.com/Papers/RS4/Papers/RS4_1006P_Black.pdf, свободный. Яз. англ. (дата обращения 16.12.2015).
4. Дегтярев А.Р., Киселев С.К. Реализация Avionics Full-Duplex Switched Ethernet для интегрированной модульной авионики с использованием COTS-компонентов // Авиационные приборы и измерительно-вычислительные комплексы: сб. науч. тр. Всероссийской научно-технической конференции ИВК-2013 / под ред. В.В. Родионова. – Ульяновск : УлГТУ, 2013. – 329 с.
5. Gamatie A., Brunette C., Delamare R., Gautier T., Talpin J. A modeling Paradigm for Integrated Modular Avionics Design [Электронный ресурс]. – Режим доступа: <https://www.irisa.fr/prive/talpin/papers/seaa06.pdf>, свободный. Яз. англ. (дата обращения 16.12.2015).
6. Oikonomou G., Phillips I., Guan L., Grigg A. An Emulator for the Network Stack of Integrated Modular Systems [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/224165588_An_emulator_for_the_network_stack_of_Integrated_Modular_Systems, свободный. Яз. англ. (дата обращения 16.12.2015).

7. Conmy P., McDermid J., High level failure analysis for Integrated Modular Avionics [Электронный ресурс]. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.9979&rep=rep1&type=pdf>, свободный. Яз. англ. (дата обращения 16.12.2015).

8. Parkinson P., Gasperoni F. High Integrity Systems Development for Integrated Modular Avionics using VxWorks and GNAT [Электронный ресурс]. – Режим доступа: http://www.adacore.com/uploads/technical-papers/HIE_Using_VxWorks_GNAT.pdf, свободный. Яз. англ. (дата обращения 16.12.2015).

9. Miller S., Cofer D., Sha L., Meseguer J. Implementing logical synchrony in Integrated Modular Avionics [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/221137269_Persistent_Logical_Synchrony, свободный. Яз. англ. (дата обращения 16.12.2015).

10. Efkeemann C., Peleska J. Model-Based testing for the Second Generation of Integrated Modular Avionics [Электронный ресурс]. – Режим доступа: http://www.informatik.unibremen.de/agbs/jp/papers/peleska_efkeemann_icstw2011.pdf, свободный. Яз. англ. (дата обращения 16.12.2015).

11. Lee Y., Kim D. Resource Scheduling in Dependable Integrated Modular Avionics [Электронный ресурс]. – Режим доступа: http://www.informatik.unibremen.de/agbs/jp/papers/peleska_efkeemann_icstw2011.pdf, свободный. Яз. англ. (дата обращения 16.12.2015).

12. Parkinson P., Kinnan L., Safety-Critical Software Development for Integrated Modular Avionics [Электронный ресурс]. – Режим доступа: http://www.vxworks.ru/rus_safety-critical-sw-dev_wp-1107.pdf, свободный. Яз. англ. (дата обращения 16.12.2015).

13. Gangkofer M., Kader D., Klockner D., White C., Transitioning to Integrated Modular Avionics with a Mission Management System [Электронный ресурс]. – Режим доступа: <http://www.dtic.mil/dtic/tr/fulltext/u2/p010970.pdf>, свободный. Яз. англ. (дата обращения 16.12.2015).

14. Tagawa G., de Souza M. An overview of the Integrated Modular Avionics Concept [Электронный ресурс]. – Режим доступа: <http://www.sbmac.org.br/dincon/2011/files/articles/071.pdf>, свободный. Яз. англ. (дата обращения 16.12.2015).
15. Бэкон Дж., Харрис Т. Операционные системы. – СПб. : Питер, 2004. – 800 с.: ил.
16. Таненбаум Э. Современные операционные системы. 3-е изд. – СПб. : Питер, 2010. – 1120 с.: ил.
17. Таненбаум, Э. С., Вудхалл А.С. Операционные системы: разработка и реализация. – СПб. : Питер, 2007. – 704 с.
18. Гатчин Ю.А., Жаринов И.О. Основы проектирования вычислительных систем интегрированной модульной авионики: монография. – М.: Машиностроение, 2010. – 224 с.
19. Жаринов И.О. Принципы построения и методы автоматизации проектирования вычислительных систем интегрированных комплексов бортового оборудования : Дисс. ... д-ра техн. наук : 05.13.12. – СПб: НИУ ИТМО. – 2011. – 298 с.
20. Жаринов О. О., Видин Б. В., Шек-Иовсепянц Р. А. Принципы построения ядра бортовой многопроцессорной вычислительной системы для авионики пятого поколения // Научно-технический вестник СПбГУ ИТМО. – 2010. – № 4 (68). – С. 21–27.
21. Гатчин Ю.А., Жаринов И.О., Жаринов О.О. Архитектура программного обеспечения автоматизированного рабочего места разработчика бортового авиационного оборудования // Научно-технический вестник информационных технологий, механики и оптики. – 2012. – № 2 (78). – С. 140–141.
22. Видин Б. В., Жаринов И. О., Жаринов О.О. Декомпозиционные методы в задачах распределения вычислительных ресурсов многомашинных комплексов бортовой авионики // Информационно-управляющие системы. – 2010. – № 1. – С. 2–5.
23. Парамонов П.П., Бобцов А.А., Видин Б. В., Жаринов И.О., Жаринов О.О., Сабо Ю.И., Шек-Иовсепянц Р. А. Проектирование систем бортового информаци-

онного обмена и их функциональных элементов: монография. – Тула: Гриф и К., 2010. – 208 с.

24. Джанджгава Г.И., Герасимов Г. И., Петкевичус П.Ю. Развитие интегрированных комплексов бортового оборудования навигации, управления и наведения летательных аппаратов в разработках Раменского приборостроительного конструкторского бюро // Авиакосмическое приборостроение. – 2008. – № 2. – С. 3–8.

25. Джанджгава Г.И., Рогалев А.П., Бабиченко А.В., Сухоруков С. Я. Интегрированная динамически реконфигурируемая система комплексной обработки информации бортовых комплексов навигации, управления и наведения // Авиакосмическое приборостроение. – 2002. – № 6. – С. 8–14.

26. Джанджгава Г.И. Авионика пятого поколения: новые задачи – новая структура // Вестник авиации и космонавтики. – 2001. – № 5. – С. 8–10.

27. Джанджгава Г.И., Герасимов Г.И., Рогалев А. П., Шерман В. М., Сухоруков С. Я., Вершков В. В. Концепция создания интегрированных комплексов бортового оборудования летательных аппаратов нового поколения // Авиакосмическое приборостроение. – 2002. – № 6. – С. 38.

28. Евгенов А. В. Направления развития интегрированных комплексов бортового оборудования самолетов гражданской авиации // Авиакосмическое приборостроение. – 2003. – № 3. – С. 48–53.

29. Турчак А. А. Архитектура вычислительных систем для интегрированной модульной авионики перспективных летательных аппаратов // Радиотехника . – 2001. – № 8. – С. 87–95.

30. Богатырев В.А., Голубев И. Ю., Беззубов В. Ф. Организация межмашинного обмена в дублированных вычислительных комплексах // Изв. вузов. Приборостроение. – 2012. – Т. 55. – № 3. – С. 8–13.

31. Богатырев В.А. К распределению функциональных ресурсов в отказоустойчивых многомашинных вычислительных системах // Приборы и системы. Управление, контроль, диагностика. – 2001. – № 12. – С. 1–5.

32. Афраимович Л. Г., Прилуцкий М. Х. Многоиндексные задачи распределения ресурсов в иерархических системах // Автоматика и телемеханика. – 2006. – № 6. – С. 194–205.
33. Буздалов Д.В., Зеленов С.В., Корныхин Е.В., Петренко А.К., Страх А.В., Угненко А.А., Хорошилов А.В. Инструментальные средства проектирования систем интегрированной модульной авионики // Труды института системного программирования РАН. – 2014. – Т. 26. – № 1. – С. 201–230.
34. Федосов Е .А . Российский проект создания нового поколения интегрированной модульной авионики с открытой архитектурой . Состояние и перспективы // Фазотрон. – 2011. – № 2. [Электронный ресурс] <http://www.media-phazotron.ru/?p=192>. (дата обращения 15.12.2015)
35. Реконфигурируемые мультиконвейерные вычислительные структуры / И.А. Каляев [и др.] – Ростов-на-Дону : Издательство ЮНЦ РАН, 2008. – 393 с.
36. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М. : Янус-К, 2003. – 380 с.
37. Семейство базовых модулей для построения реконфигурируемых многопроцессорных вычислительных систем со структурно-процедурной организацией вычислений / И.В. Беседин [и др.] // Научный сервис в сети Интернет: технологии распределенных вычислений : матер. всерос. науч. конф. – М. : Издательство Московского университета, 2006. – С. 47–49.
38. Чуюнов Г.А., Косьянчук В.В., Сельвесюк Н.И. Перспективы развития комплексов бортового оборудования на базе интегрированной модульной авионики // Известия ЮФУ. Технические науки. – 2013. – № 3. – С. 55–62.
39. Е. Федосов, В. Косьянчук, Н. Севельсюк. Интегрированная модульная авионика // Радиоэлектронные технологии. – 2015. – № 1. – С. 66–71.
40. Е.В. Книга, И.О. Жаринов, А.В. Богданов, П.С. Виноградов. Принципы организации архитектуры перспективных бортовых цифровых вычислительных систем в авионике // Научно-технический вестник информационных технологий, механики и оптики. – 2013. – № 2. – С. 163–165.

41. Абросимов О.В., Егоров К.А., Итенберг И.И., Ковальчученко А.Ф., Куликов Д.А., Сивцов С.А., Федосов Е.А., Чуянов Г.А. Базовая несущая конструкция платформы интегрированной модульной авионики. Патент на полезную модель №106404 U1 RU, МПК G06F 1/16. №2010129389/08. Заявл. 15.07.2010. Оpubл. 10.07.2011.

42. Богданов А.В., Васильев Г.А., Виноградов П.С., Егоров К.А., Зайченко А.Н., Ковернинский И.В., Петухов В.И., Романов А.Н., Смирнов Е.В., Уткин Б.В., Федосов Е.А., Шукалов А.В. Платформа интегрированной модульной авионики. Патент на полезную модель №108868 U1 RU, МПК G06F 9/00, №2011121962/08. Заявл. 01.06.2011. Оpubл. 27.09.2011.

43. Герлих Х. Модульная система авионики самолета. Патент №2413655 C2 RU, МПК B64C 19/00. №2008123940/11. Заявл. 16.11.2006. Оpubл. 10.03.2011. Бюл. № 7.

44. Егоров К.А., Итенберг И.И., Ковернинский И.В., Тимченко А.П., Федосов Е.А., Чуянов Г.А. Платформа интегрированной модульной авионики. Патент №2413280 C1 RU, МПК G06F 9/02. №2009127190/08. Заявл. 14.07.2009. Оpubл. 27.02.2011. Бюл. № 6.

45. Егоров К.А., Итенберг И.И., Ковернинский И.В., Тимченко А.П., Федосов Е.А., Чуянов Г.А. Платформа интегрированной модульной авионики. Патент на полезную модель №88462 C1 RU, МПК G06F 9/00. №2009127040/22. Заявл. 14.07.2009. Оpubл. 10.11.2009.

46. Кучерявый А.А. Современная интегрированная модульная авионика: состояние и тенденции развития. Аналитический обзор. – Ульяновск : ОАО «УКБП», 2006. – 105 с.

47. Spitzer C. R. The Avionics handbook. – Washington, DC : CRC Press, 2001. – 542 p.

48. Павлов А. М. Принцип организации бортовых вычислительных систем перспективных летательных аппаратов // Мир компьютерной автоматизации. – 2001. – № 4 [Электронный ресурс]. – Режим доступа: www.mka.ru/?p=41177, свободный. Яз рус. (дата обращения 21.11.2015).

49. Пятницких А. Бортовые компьютеры: варианты построения готовых систем // Современные технологии автоматизации. – 2008. – № 2. – С. 20–24.

50. Джонсон К., Леру П. Использование технологии объединения ресурсов для создания безопасных отказоустойчивых военных систем // Современные технологии автоматизации. – 2007. – № 4. – С. 72–76.

51. Буравлев А., Чельдиев М., Барыбин А., Костенко В., Тумакин Д., Петров Г. Масштабируемые мультипроцессорные вычислительные системы высокой производительности // Современные технологии автоматизации. – 2009. – № 3. – С. 72–82.

52. Платформа интегральной модульной авионики [Электронный ресурс]. – Режим доступа: <http://electroautomatica.ru/prodb.php?> , свободный. Яз рус. (дата обращения 10.12.2015).

53. Бортовая центральная вычислительная система БЦВС-1 [Электронный ресурс]. – Режим доступа: http://www.rpkb.ru/lines-of-business/electronic_direction/onboard-computers/onboard-central-computer-system/#bcvs-1, свободный. Яз рус. (дата обращения 10.12.2015).

54. Дегтярев А.Р. Интегрированная модульная авионика как мультипроцессорная система // Информатика и вычислительная техника: сб. науч. тр. 6-й Всероссийской научно-технической конференции аспирантов, студентов и молодых ученых ИВТ-2014 / под общ. ред. В.Н. Негоды. – Ульяновск : УлГТУ, 2014. – 502 с.

55. Дегтярев А.Р., Киселев С.К. Особенности построения реконфигурирующихся комплексов бортового оборудования // Сборник материалов Всероссийской молодежной школы семинара «Актуальные проблемы информационных технологий, электроники и радиотехники - 2015» (ИТЭР – 2015). Том 2. – Таганрог : изд-во НОЦ ЗИС КТ Южного федерального университета, 2015. – 260 с.

56. Timmerman M., Beneden B., Uhres L. RTOS Evaluation Kick Off! // Real-Time Magazine. – 1998. - № 3. – pp. 6-10.

57. Операционные системы реального времени для авионики: обзор [Электронный ресурс]. – Режим доступа:

http://rnd.cnews.ru/reviews/index_science.shtml?2008/05/05/299461_1, свободный. Яз рус. (дата обращения 15.12.2015).

58. Руководство Р-297 по вопросам разработки и квалификации интегрированной модульной авионики – М. : ОАО Авиаиздат, 2015. – 123 с.

59. Квалификационные требования КТ-178В. Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники – М. : НИИАО, 2002. – 105 с.

60. Руководство по гарантии конструирования бортовой электронной аппаратуры КТ-254 – М. : ОАО Авиаиздат, 2011. – 86 с.

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

62. Руководство 4754 по процессам сертификации высокоинтегрированных сложных бортовых систем воздушных судов гражданской авиации – М. : ОАО Авиаиздат, 2010. – 76 с.

63. Парамонов П. П., Жаринов И.О. Интегрированные бортовые вычислительные системы: обзор современного состояния и анализ перспектив развития в авиационном приборостроении // Научно-технический вестник информационных технологий, механики и оптики. – 2013. – № 2. – С. 1–17.

64. Терсков В.А.. Моделирование и оптимизация структуры многопроцессорных вычислительных систем автоматизированного управления в реальном времени : Дисс. ... д-ра техн. наук : 05.13.06. – Красноярск: САА. – 2001. – 318 с.

65. А.В. Лысенко, Г.В. Кремез. Многопроцессорная перестраиваемая бортовая вычислительная система с блоком управления реконфигурацией // Современные проблемы науки и образования – 2012. – № 4. [Электронный ресурс]. – Режим доступа: www.science-education.ru/ru/article/view?id=6787, свободный. Яз рус. (дата обращения 07.05.2016).

66. Гроль В.В., Романкевич В.А., Мораведж С.М. Об оптимизации процедуры реконфигурирования в многопроцессорных системах // Изв. Сарат. ун-та. Сер. математика. Механика. Информатика – 2012. – Т. 12. – № 4. – С. 112–115.

67. Тарасов А.А. Стратегии функциональной перестройки отказоустойчивых информационных систем при различных видах деградации // Безопасность информационных технологий – 2012. – № 2. – С. 22–31.

68. Тарасов А.А. Методы функциональной реконфигурации отказоустойчивых систем // Надежность – 2002. – № 2. – С. 29–35.

69. Чербаков Ю.В. Системный подход к прецизионным исследованиям. – СПб. : Изд-во Санкт-Петербургского университета, 1992. – 167 с.

70. Жаринов И.О., Жаринов О.О. Бортовое радиоэлектронное оборудование пилотируемых летательных аппаратов. Проектные решения. Системный анализ. Многокритериальная оптимизация: Учебное пособие. – СПб. : ГУ ИТМО, 2008. – 52 с.

71. Белов В.П., Белов В.В. Структура информационных технологий обеспечения аналитической деятельности // Проблемы автоматизации. – 1999. – № 2. – С. 12–15.

72. Герасимов Б.И. Основы теории системного анализа: качество и выбор : учебное пособие / Б.И. Герасимов, Г.Л. Попова, Н.В. Злобина. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2011. – 80 с.

73. Апполов О.Г. Теория систем и системный анализ. Курс лекций. – Уфа, 2012. – 274 с.

74. Шенброт И.М., Алиев В.М. Проектирование вычислительных систем распределенных АСУ ТП. – М. : Энергоатомиздат, 1989. – 88 с.

75. Дегтярев А.Р. Перспективные динамически реконфигурирующиеся комплексы бортового оборудования на основе интегрированной модульной авионики // Научные чтения по авиации, посвященные памяти Н.Е. Жуковского – 2015. – №3. – С. 371–375.

76. Дегтярев А.Р., Киселев С.К. Алгоритм оптимизации схемотехнического и конструкторского решения многопроцессорной системы интегрированного комплекса бортового оборудования // Вузовская наука в современных условиях: сб. матер. 49-й науч.-техн. конф. В 3 ч. Ч 2. – Ульяновск : УлГТУ, 2015. – 283 с.

77. Майника Э. Алгоритмы оптимизации на сетях и графах: пер. с англ. – М. : Мир, 1981. – 323 с., ил.
78. Таха Хемди А. Введение в исследование операций. 6-е издание. : Пер. с англ. – М. : Издательский дом “Вильямс”, 2001. – 912 с. : ил. – Парал. тит. англ.
79. Приближенные алгоритмы для NP-трудных задач : учеб.-метод. пособие / А.В. Кононов, П.А. Кононова ; Новосиб. гос. ун-т. – Новосибирск: РИЦ НГУ, 2014. -117 с.
80. Построение и анализ вычислительных алгоритмов / А. Ахо [и др.] – М. : Мир, 1979 – 536 с.
81. Ху Т.Ч., Шинг М.Т. Комбинаторные алгоритмы. – Нижний Новгород : Изд-во нижегородского гос. ун-та им. Н.И. Лобачевского, 2004. – 330 с.
82. Псиола В.В. О приближенном решении 3-х мерной задачи об упаковке на основе эвристик // Интеллектуальные системы. – 2007. – № 11. – С. 83–100.
83. **Дегтярев А.Р., Медведев Г.В. Алгоритм распределения задач в многопроцессорных комплексах интегрированной модульной авионики // Автоматизация процессов управления. – 2014. – Т. 35. – № 1. – С. 79–84.**
84. **Дегтярев А.Р., Киселев С.К. Смешанная реконфигурация в многопроцессорных комплексах интегрированной модульной авионики // Электротехнические комплексы и системы. – 2015. – Т. 11. – № 3. – С. 5–11.**
85. Дегтярев А.Р. Киселев С.К. Разработка перспективных динамически реконфигурируемых комплексов бортового оборудования // III Всероссийский конгресс молодых ученых. 8-11 апреля 2014 года: Сборник тезисов. – СПб: Университет ИТМО, 2014.
86. Дегтярев А.Р. Киселев С.К. Алгоритмы динамической реконфигурации комплексов бортового оборудования интегрированной модульной авионики // IV Всероссийский конгресс молодых ученых. 7-10 апреля 2015 года: Сборник тезисов. – СПб: Университет ИТМО, 2015.
87. Дегтярев А.Р., Киселев С.К. Подходы к оценке надежности реконфигурируемых многопроцессорных комплексов интегрированной модульной авионики // Прикладные информационные системы: вторая Всероссийская НПК (г. Улья-

новск, 25 мая – 7 июня 2015 г.): сб. науч. тр. / под. ред. Е.Н. Эгова. – Ульяновск : УлГТУ, 2015. – 533 с.

88. Дегтярев А.Р. Киселев С.К. Надежность реконфигурирующихся комплексов интегрированной модульной авионики // Автоматизация процессов управления. – 2016. – Т. 43. – № 1. – С. 25–30.

89. Дегтярев А.Р. Оценка безопасности реконфигурирующихся комплексов бортового оборудования интегрированной модульной авионики // V Всероссийский конгресс молодых ученых. 12-15 апреля 2016 года: Сборник тезисов. – СПб: Университет ИТМО, 2016.

90. Авиационные правила. Часть 25. Нормы летной годности самолетов транспортной категории – М. : ОАО Авиаиздат, 2015. – 290 с.

91. Дегтярев А.Р. Киселев С.К. Отказоустойчивые реконфигурирующиеся комплексы интегрированной модульной авионики // Электротехнические комплексы и системы. – 2016. – Т. 12. – № 1. – С. 89–99.

92. Дегтярев А.Р., Киселев С.К. Анализ деревьев неисправностей при оценке надежности реконфигурируемых мультипроцессорных систем // Радиоэлектронная техника : межвузовский сб. науч. тр. / под ред. В.А. Сергеева. – Ульяновск : УлГТУ, 2015. – 290 с.

93. Руководящий материал РМ 25-2007. Методика расчета показателей надежности (безотказности) изделий разработки ОАО «УКБП» – Ульяновск : ОАО «УКБП», 2007. – 23 с.

94. ГОСТ 27.301-95 Надежность в технике. Расчет надежности. Основные положения.

95. Тихоненков В.А., Мишин В.А. Конструирование и надежность ИВК летательных аппаратов: Учебное пособие. – Ульяновск: УлГТУ, 2002. – 298 с.

96. План управления конфигурацией курсовертикали КВ-1 КИВШ.402138.009-07. - Ульяновск : АО «УКБП», 2015. – 31 с.

СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА

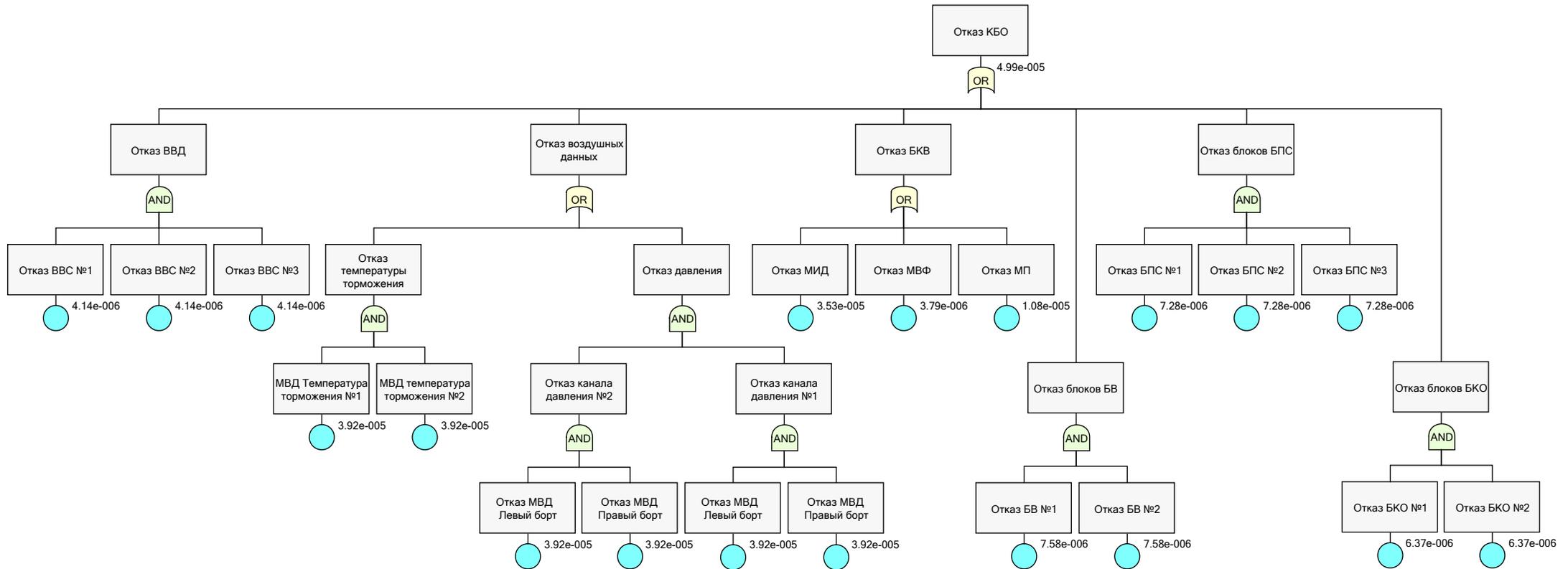
Рисунок 1.1 – Крейт системы AIMS_____	13
Рисунок 1.2 – Вычислитель и модуль формата 3U самолета A380_____	15
Рисунок 1.3 – Архитектура КБО IAS самолета F-22 Raptor _____	16
Рисунок 1.4 – Крейт 6U 10M производства ФГУП «СПб ОКБ «Электроавтоматика»_____	20
Рисунок 1.5 – КФМ крейта 6U 10M_____	21
Рисунок 1.6 – Внешний вид БЦВС-1_____	22
Рисунок 1.7 – КФМ крейта БЦВС-1_____	23
Рисунок 1.8 – Типы многопроцессорных систем_____	25
Рисунок 1.9 – Четыре ВМ, использующие общий код ОСРВ_____	27
Рисунок 1.10 – Крейт, построенный по модели “master-slave”_____	27
Рисунок 1.11 – Крейт, построенный на основе модели SMP_____	28
Рисунок 1.12 – Структура системы реального времени_____	30
Рисунок 1.13 – Структура монолитной ОС_____	31
Рисунок 1.14 – Структура уровневой ОС _____	32
Рисунок 1.15 – Структура микроядерной ОС_____	32
Рисунок 1.16 – Структура реконфигурируемой системы на основе ПЛИС____	45
Рисунок 2.1 – Системный подход при проектировании реконфигурирующегося КБО ИМА_____	52
Рисунок 2.2 – Многомерная суперповерхность реконфигурирующегося КБО ИМА_____	55
Рисунок 2.3 – Аппаратная структура КБО ИМА_____	57
Рисунок 2.4 – Функциональный граф КБО вертолета _____	60
Рисунок 2.5 – Математическая модель КБО ИМА в виде двудольного графа_____	71
Рисунок 2.6 – Функциональный граф подсистемы КБО ИМА_____	73
Рисунок 2.7 – Граф аппаратной структуры КБО ИМА с распределенными функциональными задачами_____	76

Рисунок 2.8 – Блок-схема алгоритма распределения по критерию минимума загрузки сети	77
Рисунок 3.1 – Комбинированный принцип размещения функций контроля	85
Рисунок 3.2 – Блок-схема алгоритма принятия решения о реконфигурации	90
Рисунок 3.3 – Функциональный граф подсистемы КБО вертолета с назначенными параметрами критичности	92
Рисунок 3.4 – Блок-схема алгоритма аппаратной реконфигурации КБО ИМА	97
Рисунок 3.5 – Граф G_S после работы алгоритма аппаратной реконфигурации	98
Рисунок 3.6 – Функциональные графы режимов работы КБО	104
Рисунок 3.7 – Блок-схема алгоритма смешанной реконфигурации КБО ИМА	109
Рисунок 3.8 – Результат работы алгоритма смешанной реконфигурации КБО	110
Рисунок 4.1 – Зависимость количества VM крейта от числа приложений при распределении по оптимальному алгоритму	115
Рисунок 4.2 – Зависимость количества VM крейта от показателя средней сложности приложений при распределении по оптимальному алгоритму	115
Рисунок 4.3 – Параметрическая поверхность КБО, полученная в результате работы алгоритма оптимального распределения	116
Рисунок 4.4 – Зависимость допустимого количества отказавших VM от количества функций с высокими уровнями критичности	117
Рисунок 4.5 – Обобщенная схема режимов работы реконфигурирующегося КБО ИМА	120
Рисунок 4.6 – Дерево неисправностей блока федеративной архитектуры	123
Рисунок 4.7 – Дерево неисправностей крейта архитектуры ИМА	123
Рисунок 4.8 – Блок типа БПС производства АО «УКБП»	125
Рисунок 4.9 – Блок БКВ производства АО «УКБП» и датчик магнитного курса ДМК	126

Рисунок 4.10 – Функциональные узлы, выполняющие функции приема и преобразования воздушных сигналов производства АО «УКБП»_____	126
Рисунок 4.11 – Модуль МИД разработки АО «УКБП» _____	130
Рисунок 4.12 – Модуль типа МВФ разработки АО «УКБП», применяющийся в качестве вычислителя в КБО ИМА_____	130
Рисунок 4.13 – Схема процесса определения конфигурации КБО ИМА _____	137

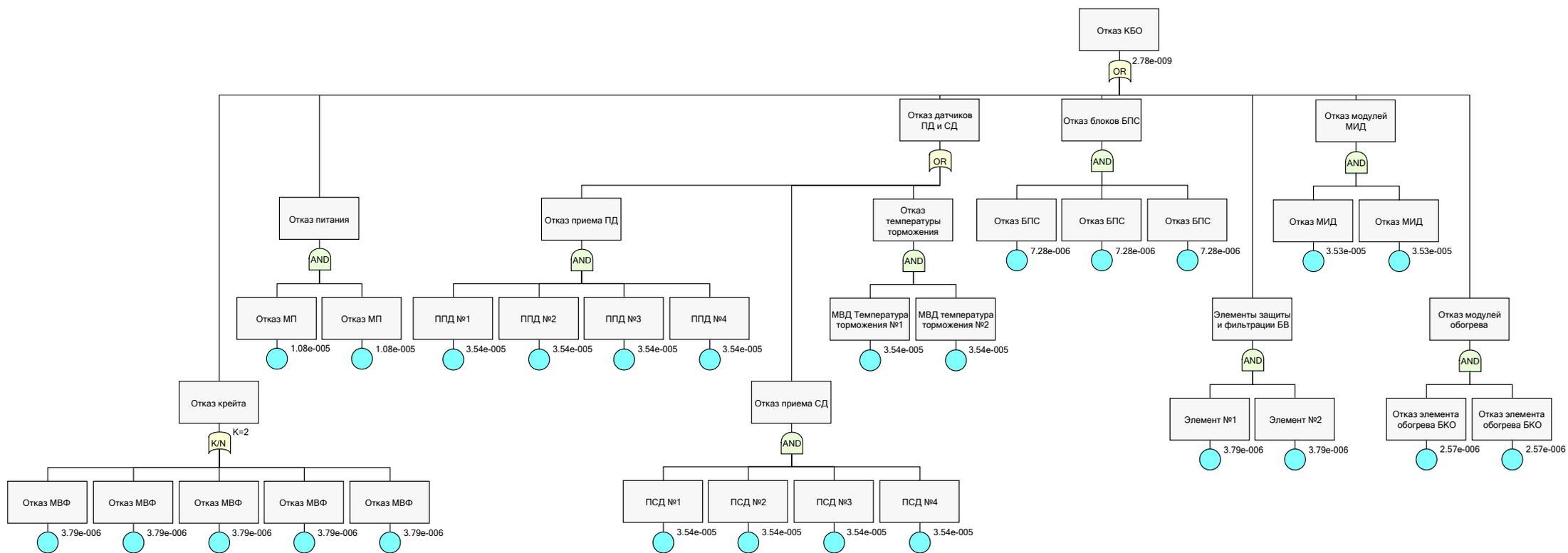
Приложение А

Дерево неисправностей распределенной системы ИМА



Приложение Б

Дерево неисправностей реконфигурирующей системы ИМА



Приложение В

УТВЕРЖДАЮ

Генеральный директор
АО «Ульяновское конструкторское
бюро приборостроения»
Д.В. Мартыненко
«26» _____ 2016г.

АКТ

внедрения результатов диссертационной работы
Дегтярева Алексея Робертовича, выполненной на тему
«Модели и алгоритмы управления конфигурацией систем
интегрированной модульной авионики» в разработках
АО «Ульяновское конструкторское бюро приборостроения»

Настоящий акт подтверждает, что ряд результатов, полученных в диссертационной работе Дегтярева А.Р., реализован при разработке платформы ИМА ИМ-21-2 для многофункционального индикатора МФИ, в частности:

- была использована математическая модель аппаратной и программной составляющих платформы ИМА ИМ-21-2;
- был использован алгоритм распределения программных приложений по минимуму загрузки сети передачи данных и соответствующая ему параметрическая поверхность вариантов построения реконфигурирующегося крейта ИМА для анализа реальных функциональных приложений, запускаемых на платформе ИМА ИМ-21-2;
- были задействованы алгоритмы изменения конфигурации системы ИМА и механизм их реализации через определение таблиц реконфигурации для различных режимов полета и технического состояния;
- была рассмотрена обобщенная методика предварительного определения аппаратной конфигурации ИМА, с последующей реализацией реконфигурирующейся системы применительно к платформе ИМА ИМ-21-2.

Использование научно обоснованных технических решений по методологии проектирования КБО позволило повысить эффективность функционирования разрабатываемых систем, сократить сроки и снизить затраты на разработку платформы интегрированной модульной авионики.

Главный конструктор

А.Б. Виноградов

Директор программ по оборудованию
гражданских, транспортных и «тяжелых»
военных самолетов

Д.Л. Федоров