ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

Кафедра вычислительных машин, комплексов, систем и сетей

Н.И. Романчева

МЕТОДЫ И СРЕДСТВА ДИАГНОСТИКИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ В ГРАЖДАНСКОЙ АВИАЦИИ

Учебное пособие

Утверждено редакционно-издательским советом МГТУ ГА в качестве учебного пособия

Москва
ИД Академии Жуковского
2020

УДК 681.324
ББК 6Ф6.5
Р69

Печатается по решению редакционно-издательского совета
Московского государственного технического университета ГА

Рецензенты:
Терентьев А.И. (МГТУ ГА) – канд. техн. наук, доцент;
Юркевич Е.В. (гл. сотрудник ИПУ им. Трапезникова РАН) – д-р техн. наук, профессор

Романчева Н.И.

Р69 Методы и средства диагностики вычислительных систем и сетей в гражданской авиации [Текст] : учебное пособие / Н.И. Романчева. – М. : ИД Академии Жуковского, 2020. – 80 с., 15 ил., лит.: 13 наим.

ISBN 978-5-907275-20-1

Учебное пособие содержит базовый лекционный материал по дисциплине «Методы и средства диагностики вычислительных систем и сетей в гражданской авиации». В учебном пособии рассматривается комплекс вопросов, связанных с формированием у аспирантов профессиональных компетенций – представлений, умений и знаний теоретических основ, методов алгоритмов контроля, диагностирования, повышения эффективности, надежности и живучести АСУТП, АСУП, АСТПП на этапах разработки, внедрения и эксплуатации. Приводятся контрольные вопросы.

Учебное пособие издается в соответствии с учебным планом для аспирантов по направлению подготовки 09.06.01 «Информатика и вычислительная техника», направлению 05.13.06 «Автоматизация и управление технологическими процессами и производством (транспорт)» очной формы обучения.

Рассмотрено и одобрено на заседании кафедры 25.02.2020 г. и методического совета 25.02.2020 г.

УДК 681.324
ББК 6Ф6.5
Св. тем. план 2020 г.
поз. 27

РОМАНЧЕВА Нина Ивановна
МЕТОДЫ И СРЕДСТВА ДИАГНОСТИКИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ В ГРАЖДАНСКОЙ АВИАЦИИ

Учебное пособие

В авторской редакции

Подписано в печать 29.06.2020 г.
Формат 60x84/16 Печ. л. 5 Усл. печ. л. 4,65
Заказ № 594/0413-УП01 Тираж 15 экз.

Московский государственный технический университет ГА
125993, Москва, Кронштадтский бульвар, д. 20

Издательский дом Академии имени Н. Е. Жуковского
125167, Москва, 8-го Марта 4-я ул., д. 6А
Тел.: (495) 973-45-68 E-mail: zakaz@itsbook.ru

ISBN 978-5-907275-20-1

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Настоящее учебное пособие строится на материале, который читается для аспирантов 2-го года направления подготовки 09.06.01 «Информатика и вычислительная техника», направленность 05.13.06 Автоматизация и управление технологическими процессами и производством (транспорт) очной формы обучения «Методы и средства диагностики вычислительных систем и сетей в гражданской авиации» по дисциплине ДВ1.3 «Методы и средства диагностики вычислительных систем и сетей в ГА».

Термин “диагностика” происходит от греческого слова diagnossis, что означает распознавание, определение. Термин “техническая диагностика” стал активно употребляться в литературе с середины 60-х г.г. XX века. В настоящее время диагностированию технических объектов уделяется большое внимание как средству существенного повышения их надежности. Это объясняется тем, что разработка и внедрение в практику эксплуатации различных технических средств, методов и средств их диагностирования позволяет повысить безотказность, ремонтопригодность и долговечность средств вычислительной техники (СВТ), вычислительных систем и сетей, предупреждать аварии, прогнозировать остаточный ресурс и значительно увеличить надежность и экономичность вышеперечисленных объектов.

Организация систем диагностирования таких сложных систем, как вычислительные системы и сети, вычислительные комплексы, различного рода сложные вычислительные устройства, АСУ ТП – весьма сложная проблема. С помощью методов математического моделирования можно сравнивать и выбирать рациональные структуры систем диагностирования на ранних стадиях их создания до непосредственной разработки и тем самым сокращать сроки проектирования, снижать затраты материальных ресурсов и способствовать получению эффективных решений.

Под термином вычислительная система (ВС) будем понимать совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенная для сбора, хранения, обработки и распределения информации. Создание ВС преследует следующие основные цели: повышение производительности системы и надежности и достоверности вычислений; предоставление пользователям дополнительных сервисных услуг и т.д.

Отличительной особенностью ВС является наличие в ней нескольких вычислителей, реализующих параллельную обработку. Параллелизм в вычислениях в значительной степени усложняет управление вычислительным процессом, использование технических и программных ресурсов. Параллелизм выполнения операций существенно повышает быстродействие системы, значительно повысить и надежность (при отказе одного компонента системы его функции может взять на себя другой), и достоверность функционирования системы, если операции будут дублироваться, а результаты их выполнения сравниваться.

Возрастающая сложность вычислительных систем, в том числе используемых в гражданской авиации, необходимость совершенствования как качественных, так и количественных характеристик, приводит к необходимости разработки средств диагностирования как с использованием традиционных средств автоматизации диагностики на всех этапах вычислительной системы с целью улучшения характеристик по быстродействию, надежности функционирования, так и с применением методов прогнозной аналитики, базирующихся на AI (Artificial intelligence) и позволяющим адаптироваться к задаваемым параметрам.

Автоматизация процесса проверки и наладки сложных технических систем – важнейшее средство ускорения процесса создания, выпуска, внедрения и повышения эффективности создаваемых систем различного назначения. Наиболее выгодной является интегрированная автоматизация процессов проверок технических систем, включающая в себя автоматизацию процессов подготовки проверочных воздействий, организацию самого процесса проверки и принятия решения, устранение неисправностей, прогноз состояния объекта. Поэтому важное значение приобретает теория рациональной организации процессов проверки, включающая в себя анализ моделей объектов диагноза, выбор технических средств для проверки и организация их взаимодействия, увязку их с процессами восстановления отказавших элементов.

В данном учебном пособии рассматриваются основы диагностики и теории надежности, задачи, методы диагностики вычислительных систем и сетей и приемы диагностирования элементов. Изложены статистические методы распознавания и разделения в пространстве признаков, метрические и логические методы диагностики и ее приложение к задачам диагностики. Материал изложен в 3-х разделах. В первом разделе рассмотрены основные понятия и методы оценки надежности, понятие технической эксплуатации вычислительных систем и сетей. Особое внимание уделено системам, используемым в гражданской авиации. Во втором разделе приведена классификация систем контроля вычислительных систем и сетей. Рассмотрены особенности аппаратных и программных систем контроля. В третьем разделе рассматриваются системы диагностики вычислительных систем и сетей, состав и показатели качества систем диагностики. Приводится классификация и описание методов тестового диагностирования и синтеза диагностических тестов. Также рассматриваются модели диагностируемых устройств. Для контроля качества усвоения материала каждый раздел завершается перечнем контрольных вопросов для самопроверки.

Материалы курса «Методы и средства диагностики вычислительных систем и сетей в гражданской авиации» являются обработанными и оптимизированными материалами из открытых источников, открытых материалов ведущих предприятий в области диагностики ВС.

РАЗДЕЛ 1. НАДЕЖНОСТЬ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

1.1. Основные понятия

Увеличивающаяся сложность вычислительных систем и сетей, используемых для решения задач в области гражданской авиации, требует внимательного отношения к вопросам обеспечения их надежности на всех стадиях жизни, начиная от разработки и заканчивая эксплуатацией.

С точки зрения технологического аспекта использование средств вычислительной техники в вычислительных системах и сетях и обеспечение на этой основе автоматизации решения каких-либо задач проявляется в близком термине: автоматизированная система – «система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций» [2]. Выполнение таких операций, как представление, хранение и обработка информации с помощью средств вычислительной техники в рамках информационных систем, является свойством, присущим автоматизированным системам.

Автоматизированные информационные системы в настоящее время являются неотъемлемой частью современного инструментария информационного обеспечения различных видов деятельности и наиболее бурно развивающейся отраслью индустрии информационных технологий, в том числе в гражданской авиации. Системы диагностирования различных технических объектов являются актуальным подклассом автоматизированных информационных систем. Если объект диагностирования (ОД) - вычислительная система или сеть (ВС), то для получения необходимой информации о техническом состоянии ОД желательно наличие в составе ВС подсистемы мониторинга [3]. В такой подсистеме должны автоматически накапливаться данные о техническом состоянии программно-аппаратных средств и об их диагностических признаках. Это обеспечит возможность постоянного развития и совершенствования системы обеспечения надежности ВС [3].

Целью диагностирования ВС является определение вида технического состояния ВС с обнаружением и локализацией различных дефектов на всех стадиях жизненного цикла и во всех компонентах ВС.

Теория надежности изучает процессы возникновения отказов объектов в различные моменты времени и способы устранения этих событий. Под объектом понимается предмет определенного целевого назначения, рассматриваемый на этапе разработки требований, проектирования, производства, эксплуатации. В качестве объектов могут выступать различные системы, в том числе вычислительные системы и сети.

В соответствии с [1] *надежность* - это свойство объекта сохранять во времени в установленных пределах значения всех параметров,

характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, хранения и транспортирования. Надежность является комплексным свойством, которое в зависимости от назначения объекта и условий его применения может включать в себя:

- *безотказность* - свойство объекта непрерывно сохранять способность выполнять требуемые функции в течение некоторого времени или наработки в заданных режимах и условиях применения;

- *ремонтопригодность* - свойство объекта, заключающееся в его приспособленности к поддержанию и восстановлению состояния, в котором объект способен выполнять требуемые функции, путем технического обслуживания и ремонта;

- *восстанавливаемость* - свойство объекта, заключающееся в его способности восстанавливаться после отказа без ремонта. Для восстановления могут требоваться или не требоваться внешние воздействия. Для случая, когда внешние воздействия не требуются, может использоваться термин *самовосстанавливаемость*;

- *долговечность* - свойство объекта, заключающееся в его способности выполнять требуемые функции в заданных режимах и условиях использования, технического обслуживания и ремонта до достижения предельного состояния;

- *сохраняемость* - свойство объекта сохранять способность к выполнению требуемых функций после хранения и (или) транспортирования при заданных сроках и условиях хранения и (или) транспортирования;

- *готовность* - свойство объекта, заключающееся в его способности находиться в состоянии, в котором он может выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания и ремонта в предположении, что все необходимые внешние ресурсы обеспечены. Готовность зависит от свойств безотказности, ремонтопригодности и восстанавливаемости объекта.

Требуемые функции и критерии их выполнения устанавливают в нормативной, конструкторской, проектной, контрактной или иной документации на объект. Критерии выполнения требуемых функций могут быть установлены, например, заданием для каждой функции набора параметров, характеризующих способность ее выполнения, и допустимых пределов изменения значений этих параметров.

Различают следующие виды технического состояния ВС: исправное и неисправное, работоспособное и неработоспособное, правильное функционирование и неправильное функционирование.

Операции, связанные с диагностированием ВС, заключаются в определении вида технического состояния - *задача контроля*; определении характера ошибки (сбой или отказ) - *задача классификации*; поиск отказавшего сменного элемента - *локализация,* устранение ошибки - *задача восстановления*.

1.2. Понятие технической эксплуатации ЭВМ. Основные эксплуатационные характеристики ЭВМ, систем и сетей

Степень пригодности средств вычислительной техники к использованию по назначению и возможности ее технического обслуживания определяются эксплуатационными характеристиками СВТ, в частности эксплуатационными характеристиками ЭВМ. К основным эксплуатационным характеристикам относятся: работоспособность, безотказность, сохранность, ремонтопригодность, долговечность, надежность, производительность.

Работоспособность - это способность СВТ функционировать, обеспечивая выполнение заданных функций с параметрами, установленными требованиями технической документации. Эта характеристика позволяет судить о состоянии оборудования в определенный момент времени. Однако при эксплуатации важно знать состояние СВТ не только в данный момент, но и способность выполнять возложенные на технику задачи в течение заданного промежутка времени. Для этих целей вводятся понятия безотказность, сохранность, ремонтопригодность, долговечность, производительность, надежность.

Безотказность - это способность сохранять работоспособность в течение заданного интервала времени при определенных условиях эксплуатации СВТ.

Сохранность – этой характеристикой пользуются на этапе хранения ЭВМ, под которой понимают способность СВТ сохранять исправное состояние при заданных условиях хранения. Ремонтопригодность – это характеристика СВТ с точки зрения приспособленности к ремонту, т.е. удобства доступа к блокам, монтажу, приспособленность оборудования к устранению неисправности и т.п. Требования к ремонтопригодности предъявляются в зависимости от условий эксплуатации СВТ. Например, некоторые периферийные устройства, а также бортовые или стационарные компьютеры, в силу своей специфики использования не рассчитаны на обычное техническое обслуживание, и поэтому относятся к неремонтопригодным.

Долговечность – это свойство СВТ сохранять работоспособность до предельного состояния с необходимыми перерывами для технического обслуживания и ремонтов. Надежность - это свойство устойчиво функционировать при заданных условиях обслуживания и эксплуатации СВТ.

Производительность - это важное понятие, характеризующее эксплуатационные свойства ЭВМ и некоторых периферийных устройств. На протяжении всего развития ЭВМ для оценки их производительности предлагались различные критерии и методы. С развитием и совершенствованием ЭВМ различных поколений сравнивать их по одному определенному критерию нельзя. Если ЭВМ первых поколений сравнивались по быстродействию (количеству команд, выполняемых в секунду), то для ЭВМ современных поколений стали вводить такие понятия, как общая производительность машины, вычислительная мощность, производительность при решении определенного вида задач и другим параметрам. В зависимости от

области применения ЭВМ менее быстродействующая машина, но имеющая лучший набор команд для решения конкретной задачи часто имеет большую производительность, чем более быстродействующая машина [3].

1.3. Аналитические методы оценки надежности

Для расчета показателей надежности технических систем применяются аналитические методы. Методы оценки надежности сложных систем могут быть сведены в следующие группы: методы, базирующиеся на аппарате теории случайных процессов; методы, использующие аппарат теории марковских и полумарковских процессов; методы, основанные на аппарате теории функции случайных аргументов. На практике используют методы имитационного и статистического моделирования (метод Монте-Карло) [4].

Теория случайных процессов служит основой аналитических методов расчета показателей надежности. Расчет надежности сложных технических систем часто базируется на предположении о том, что время безотказной работы и время восстановления элементов имеют экспоненциальные распределения вероятностей.

Процессы, протекающие в системах с экспоненциальным распределением интервалов времени, являются марковскими, т.е. при которых вероятность перехода системы в новое состояние зависит только от состояния системы в настоящий момент и не зависит от того, когда и каким образом система перешла в это состояние. При экспоненциальном распределении случайного времени пребывания системы в каждом из возможных состояний марковский процесс является однородным, т.е. интенсивности переходов между состояниями не зависят от времени. Однородные марковские процессы с конечным числом состояний и непрерывным временем являются основным математическим аппаратом исследования надежности сложных систем с восстановлением. Это объясняется тем, что именно они позволяют получать аналитические выражения или конструктивные вычислительные схемы для расчета различных показателей надежности. Кроме того, в подавляющем большинстве случаев исходными данными для элементов являются либо константные интенсивности отказов, либо средние наработки до отказа.

Построение марковских моделей надежности происходит следующим образом. Пусть объект исследования находится в некоторых состояниях, число которых конечно, равно n. На основе информации о структуре и принципах функционирования исследуемой системы определяется множество ее возможных состояний, которое разделяется на два подмножества: работоспособных состояний и состояний отказа. Из i-го состояния в j-ое объект переходит с постоянной интенсивностью λ_{ij}, обратно с постоянной интенсивностью μ_{ij}. Строится граф переходов, вершинами которого являются состояния системы, а ребрами - возможные переходы между состояниями (рис.1). Интенсивности переходов определяются характеристиками безотказности и ремонтопригодности элементов системы. Для определения

вероятностей каждого из состояний применяют систему дифференциальных уравнений А.Н.

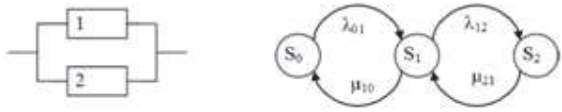

Рисунок 1 – Граф переходов

Коломогорова, решение которой позволяет получить требуемые показатели надежности [8]:

$$\begin{cases} dP_0/dt = -\lambda_{01}P_0(t) + \mu_{10}P_1(t) \\ dP_1/dt = \lambda_{01}P_0(t) - (\lambda_{12} + \mu_{10})P_1(t) + \mu_{21}P_2(t) \\ dP_2/dt = -\mu_{21}P_2(t) + \lambda_{12}P_1(t) \end{cases}$$

Получить систему уравнений можно по виду графа состояний, если пользоваться следующим правилом: для каждого из возможных состояний объекта записывается уравнение, в левой части которого dPi/dt, а справа – столько слагаемых, сколько стрелок графа соприкасается с данным состоянием. Если стрелка направлена в данное состояние, то перед слагаемым ставится плюс, иначе – минус. Каждое из слагаемых будет равно произведению интенсивности перехода из данного состояния (либо в данное состояние) на вероятность состояния, из которого выходит стрелка.

Оценка параметров надежности технических систем с использованием графов позволяет учитывать любые факторы, влияющие на систему.

Недостатком описания системы графом состояний является сложность ввода данных и методов определения характеристик надежности для систем с большим количество состояний.

Процессы, протекающие в системах с произвольными распределениями интервалов времени (Эрланга, нормальное), являются полумарковскими, т.е. при которых вероятность перехода системы из одного состояния в другое зависит от времени, проведенного в первом состоянии. Возможности применения методов, основанных на полумарковских процессах, ограничены (позволяют определять лишь стационарные значения показателей надежности), поскольку в общем виде на их основе не удается разработать математическую модель восстанавливаемой технической системы с учетом структурной избыточности и любой дисциплины ремонта.

Многомерные марковские процессы описывают функционирование технических систем при произвольных распределениях времен безотказной работы и восстановления элементов с учетом структурной и временной избыточности, с учетом контроля технических средств, с учетом нескольких видов отказов. Расчет показателей надежности методом многомерных марковских процессов осуществляется с помощью статистического моделирования, требующего огромных затрат времени и памяти ЭВМ [4].

Методы теории случайных процессов применяются для оценки надежности различных вычислительных систем и комплексов с непрерывными технологическими процессами, для исследования надежности вычислительных систем и сетей предприятий ГА. Надежность таких систем зависит от множества факторов, большинство из которых являются случайными. Модели, основанные на математическом аппарате случайных событий и марковских, полумарковских случайных процессов, позволяют учесть влияние на надежность структурного и временного резерва, ограничений и степени независимости работы и уровня контроля за состоянием основных элементов вычислительных систем и сетей.

Для оценки надежности сложных технических систем с малым числом состояний могут использоваться асимптотические методы. Установлена асимптотическая независимость показателей надежности от исходных распределений. Распределение длительности безотказной работы резервированных систем в условиях «быстрого» восстановления асимптотически экспоненциально.

Недостатком асимптотических методов, ограничивающим их применение, является локальность получаемых решений. Они позволяют найти решения задачи лишь в небольших пределах изменения параметров системы. На практике же часто нужно выйти за эти пределы [10]. Асимптотические приближенные способы расчета показателей надежности электрических систем применяются для решения проектных и эксплуатационных задач, для исследования моделей резервирования и массового обслуживания.

Для анализа надежности систем с неэкспоненциальными распределениями применяются следующие методы: логико-вероятностные, дифференциальный метод разложения на фазы, метод Кендалла, метод аппроксимации интенсивностей, графовые, экспертной оценки (укрупнения состояний), эвристические, декомпозиции (эквивалентирования), аналитико-статистические, диффузионных процессов.

Логико-вероятностные методы анализа надежности сложных технических систем используют математический аппарат бинарной алгебры логики и теорию вероятности. Методы теории массового обслуживания, к которым относятся дифференциальный метод разложения на фазы, метод Кендалла, позволяют сводить немарковскую модель к марковской. Данные методы позволяют использовать лишь распределения Эрланга и приводят к значительному увеличению числа состояний, поэтому могут использоваться для расчета стационарных характеристик надежности и вероятности безотказной работы для систем кратковременного действия [5]. Логико-вероятностный метод расчета надежности вычислительной системы и сетей с использованием дерева отказов применяется, когда число различных отказов системы относительно невелико (например, для анализа надежности автоматизированной системы диспетчерского управления). Этот метод широко

распространился при исследованиях надежности технологических систем, например, АСУТП [8].

Методы ступенчатой аппроксимации интенсивностей отказов и восстановлений элементов применяются для оценки надежности систем, имеющих незначительное число состояний и медленно изменяющиеся интенсивности (например, телекоммуникационных систем и сетей [9]).

Для прогнозирования надежности объектов применяют методы эвристического прогнозирования (экспертной оценки). Методы эвристического прогнозирования основаны на статистической обработке независимых оценок значений ожидаемых показателей надежности разрабатываемого объекта (индивидуальных прогнозов), даваемых группой квалифицированных специалистов (экспертов) на основе предоставленной им информации об объекте, условиях его эксплуатации, планируемой технологии изготовления и других данных, имеющихся в момент проведения оценки. Опрос экспертов и статистическую обработку индивидуальных прогнозов показателей надежности проводят общепринятыми при экспертной оценке любых показателей качества методами (например, методом Дельфи).

Сущность эвристического метода оценки надежности восстанавливаемых систем заключается в объединении групп элементов этой системы в один эквивалентный элемент, который характеризуется альтернирующим процессом восстановления. Тем самым происходит уменьшение числа элементов в системе. Метод не позволяет установить погрешность вычислений и применяется исключительно для случая высоконадежных элементов и систем (например, для построения высоконадежных систем объектов КИИ).

Метод декомпозиции (эквивалентирования) сложных технических систем основан на построении математических моделей, позволяющих получать достаточно точные верхнюю и нижнюю границы оцениваемого показателя надежности. Метод эквивалентирования последовательных и дублированных цепей получил широкое распространение для расчета надежности систем с большим числом элементов при параллельном и последовательном их соединении [7].

Метод статистического моделирования (или метод Монте-Карло) применяется для исследования поведения вероятностных систем в условиях, когда неизвестны в полной мере внутренние взаимодействия в этих системах. Этот метод заключается в воспроизведении исследуемого физического процесса при помощи вероятностной математической модели и вычислении характеристик этого процесса. Одно такое воспроизведение функционирования системы называют реализацией (или испытанием). После каждого испытания регистрируют совокупность параметров, характеризующих случайный исход реализации. Метод основан на многократных испытаниях построенной модели с последующей статистической обработкой полученных данных с целью определения числовых характеристик рассматриваемого процесса в виде статистических оценок его параметров. Процесс моделирования

функционирования технической системы сводится к машинной имитации изучаемого процесса, который копируется на ЭВМ со всеми сопровождающими его случайностями [10]. Метод статистического моделирования является наиболее эффективным, а в ряде случаев - единственно возможным для оценки показателей надежности уникальных или малосерийных изделий (например, оборудование атомных установок). Статическая оценка законов распределения отказов применяется для различного оборудования телекоммуникационных систем.

Методы имитационного моделирования в целом являются универсальными и допускают рассмотрение систем с большим количеством элементов. Однако их использование в качестве метода исследования задач надежности целесообразно лишь тогда, когда трудно или невозможно получить аналитическое решение. Основными этапами такого исследования являются: построение формальной модели, разработка программ имитации траекторий модели, проведение имитационных экспериментов.

При анализе высоконадежных систем с помощью имитационной модели возникают проблемы, связанные с большими затратами машинного времени, необходимого для вычислений с требуемой точностью. С увеличением надежности элементов эффективность моделирования уменьшается, и оно становится практически нереализуемым. Методы статистического и имитационного моделирования не позволяют в полном объеме определять надежность системы, если учесть большое количество сопутствующих факторов, влияющих на ее функционирование.

В теории надежности больших систем актуальной задачей является разработка математического аппарата для расчета, анализа и прогнозирования надежности функционирования, позволяющих анализировать технические системы, описываемые уравнениями больших размерностей. При разработке математической модели технической системы с большим числом состояний сталкиваются со следующими препятствиями, существенно затрудняющими анализ ее надежности: неоднозначность понятия отказа системы, взаимовлияние отказов элементов и частей системы, неопределенность исходных данных, многокритериальность, восстанавливаемость.

Для оценки показателей надежности сложных технических систем с большим числом состояний используются методы имитационного моделирования, асимптотического анализа, случайных процессов и связанных с ними интегро-дифференциальных уравнений. В теории надежности предполагается, что технические системы и их компоненты могут пребывать в двух возможных состояниях: работоспособном и состоянии отказа. При этом отказы элементов независимы, и система попадает в состояние отказа при отказе определенного числа элементов. Для сложных систем эти допущения часто бывают неприемлемыми. Между характеристиками отдельных частей системы имеется тесная взаимосвязь, и отказы отдельных частей системы являются зависимыми событиями.

Сложная техническая система является, как правило, многофункциональной. При этом количество выполняемых системой функций может достигать нескольких десятков. В реализации одной функции может участвовать большое число компонентов. Один и тот же компонент может быть задействован в выполнении нескольких функций. Поэтому компоненты, образующие систему, имеют различную длительность эксплуатации. При изучении надежности систем, выполняющих несколько функций, как правило, применяется функциональный подход, при котором описание надежности производится по каждой функции в отдельности, поэтому надежность системы характеризуется вектором показателей надежности всех ее функций.

Методы анализа надежности сложных систем должны учитывать: наличие последействия отказов вычислительных систем и сетей, а также систем с восстановлением, характер отказа элементов системы, структуру сложной системы, неодновременность работы элементов. Математические модели функционирования сложных систем с точки зрения надежности, полученные без учета перечисленных факторов, не могут быть адекватными реальным системам.

Сравнительный анализ существующих методов (оценка их возможностей) показывает, что для оценки надежности и эффективности функционирования каждой сложной технической системы с большим числом состояний необходимо, основываясь на традиционных методах, необходимо учитывать особенности ее функционирования.

1.4. Оценка надежности с использованием аналитического аппарата теории массового обслуживания

Теорию массового обслуживания рассматривают как раздел прикладной математики, изучающей процессы, связанные с удовлетворением массового спроса на выполнение какого-либо вида услуг с учетом случайного характера спроса и обслуживания. *Системой массового обслуживания* называется любая система, предназначенная для обслуживания каких-либо заявок (требований), поступающих в нее в случайные моменты времени.

Каждая система массового обслуживания (СМО) может быть представлена в виде определенного числа обслуживающих единиц, которые называются каналами обслуживания. В качестве канала могут рассматриваться различного вида приборы и приспособления, вычислительная машина, коллектив людей или отдельный исполнитель, выполняющий определенный вид работ. По числу каналов СМО делится на одноканальные и многоканальные системы.

Функционирование любой СМО заключается в обслуживании поступающего в нее потока заявок или требований. Заявки обычно поступают нерегулярно, образуя случайный поток заявок (требований). На обслуживание заявки также необходимо определенное время. Случайный характер потока заявок и времени обслуживания приводит к неравномерной загрузке СМО. В какие-то периоды времени скапливается большое количество заявок (они либо

становятся в очередь, либо покидают СМО, не получив обслуживания), в другие периоды СМО может работать с недогрузкой или простаивать.

Пример одноканальной СМО – АРМ оператора системы DCS. Пример многоканальной СМО – АРМ оператора, подключенного к нескольким системам DCS.

Система массового обслуживания может быть с отказами и с очередью. В СМО с отказами заявка, пришедшая в момент, когда все каналы заняты, получает отказ и в дальнейшем процессе работы СМО не участвует. В СМО с очередью заявка, пришедшая в момент занятости всех каналов, не покидает СМО, а становиться в очередь и ждет, пока не освободится какой либо канал. Наглядным примером СМО с очередью является работа колл-центра авиакомпании при возникновении сбоев в расписании рейсов воздушных судов, связанных, например, с погодными условиями. В зависимости от числа мест в очереди различают СМО с отказами и без отказов. В СМО с отказами число мест в очереди конечно и вследствие вероятностного характера, как входящего потока, так и процессов обслуживания, существует ненулевая вероятность того, что поступившая на вход СМО заявка застанет все каналы занятыми обслуживанием и все места в очереди занятыми ожидающими заявками, то есть она получит отказ. Примером СМО с отказами является очередь по обслуживанию потенциальных авиапассажиров перед обеденным перерывом, когда кассир, производящий обслуживание, ограничивает число людей, которые будут приняты до обеда. В СМО без отказов заявка либо сразу назначается на обслуживание, если в момент ее поступления свободен хотя бы один канал, либо, безусловно, принимается в очередь.

В зависимости от допустимого времени пребывания заявки в системе различают СМО с «нетерпеливыми» и «терпеливыми» заявками. В системе с нетерпеливыми заявками заявка может «уйти» из системы, если время пребывания ее в СМО превысит некоторое допустимое значение, которое в общем случае может быть случайным и характеризуется определенным законом распределения. Терпеливые заявки, попав в СМО, непременно дождутся конца обслуживания.

Системы массового обслуживания различаются не только по ограничению в очереди, но и по дисциплине поступающих заявок. При этом рассматриваются различные правила обслуживания заявок: обслуживается ли заявка в порядке поступления, в случайном порядке или она обслуживается вне очереди (СМО с приоритетом). В зависимости от принятых в СМО дисциплин ожидания и обслуживания различают СМО с бесприоритетными и приоритетными дисциплинами. В СМО с бесприоритетной дисциплиной обслуживания все заявки считаются равноправными, из них никто не получает каких-либо преимуществ по отношению к другим заявкам. В СМО с приоритетной дисциплиной обслуживания одни типы заявок имеют более высокий приоритет (более важны с точки зрения принятого в данной системе критерия эффективности по отношению к другим типам заявок). При таком

подходе может наблюдаться «выталкивание» заявок из очереди. Приоритетный метод устранения неисправностей, например, может быть установлен при устранении неисправностей в вычислительных системах с учетом их приоритетности для организации. Обобщенная структурная схема системы массового обслуживания представлена на рис.2 [10].

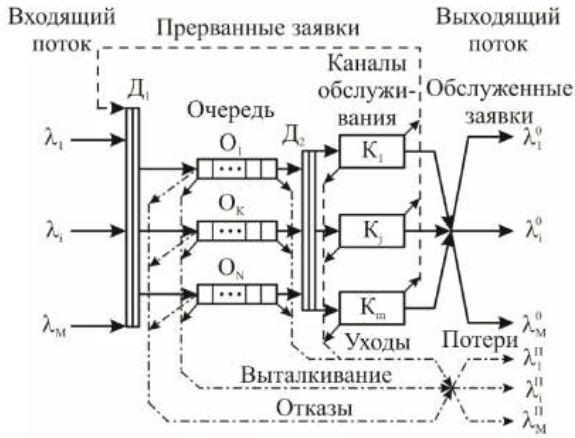

Рисунок 2 – Обобщенная структурная схема СМО

На вход СМО поступает поток заявок (требований). Потоком называется последовательность событий. Поток заявок – поток, состоящий из заявок на обслуживание. Поток заявок, нуждающихся в обслуживании и поступающих в систему, называется входящим потоком (последовательность однородных событий, поступающих через случайные интервалы времени). В качестве прим ера можно привести поток отказов оборудования вычислительной сети. *Средней интенсивностью* входящего потока называется математическое ожидание числа требований, поступающих в единицу времени. Интенсивность обслуживания характеризуется средним числом заявок, обслуживаемых в единицу времени. Эта величина обратна средней длительности обслуживания. Совокупность обслуженных и потерянных заявок образует выходящий поток СМО. В зависимости от структуры выходящего потока различают СМО с потерями и без потерь. Для СМО с потерями присуще отсутствие ограничений на число мест в очереди («терпеливые» заявки), поэтому выходящий поток будет состоять только из обслуженных заявок.

В зависимости от характера источника заявок различают разомкнутые и замкнутые СМО. В разомкнутых СМО число поступивших заявок неограниченно. В замкнутых СМО - обслуженные заявки возвращаются в источник, и через случайное время могут появиться на входе СМО.

Для простейшего потока частота поступления требований в систему подчиняется закону Пуассона, то есть вероятность поступления за время t

ровно k требований задается по формуле:

$$P_k(t) = \frac{(\lambda t)^k}{k!} \times e^{-\lambda t} \quad (1)$$

Простейший поток обладает тремя основными свойствами: ординарностью, стационарностью и отсутствием последействия. Ординарность потока означает практическую невозможность одновременного поступления двух и более требований (вероятность такого события неизмеримо мала по отношению к рассматриваемому промежутку времени, когда последний устремляют к нулю).

Важной характеристикой СМО является время обслуживания требований в системе. Время обслуживания одного требования является, как правило, случайной величиной и, следовательно, может быть описано законом распределения. Наибольшее распространение в теории, и особенно, в практических приложениях получил экспоненциальный закон распределения времени обслуживания. Функция распределения для этого закона имеет вид

$$F(t) = 1 - e^{-\mu t} \quad (2)$$

То есть вероятность того, что время обслуживания не превосходит некоторой величины *t*, определяется формулой (2), где μ — параметр экспоненциального закона распределения времени обслуживания требований в системе, то есть величина, обратная среднему времени обслуживания $\bar{t}_{об}$:

$$\mu = 1/\bar{t}_{об} \quad (3)$$

Применение аналитического метода теории массового обслуживания позволяет решить задачи планирования, оценки и оптимизации качества обслуживания заявок в ВС, в частности могут вырабатываться рекомендации по рациональному построению, организации работы и регулированию потока заявок при минимальных затратах, связанных с простоем обслуживающих каналов, в целях обеспечения конкурентоспособности и эффективности работы систем гражданской авиации.

1.5. Применение теории функций случайных аргументов для определения надежности системы

Пусть задана произвольная функция n аргументов:

$$Y = \varphi(X_1, X_2, ..., X_n)$$

Если аргументы X представляют собой случайные величины, то при конечном n функция Y также является случайной величиной и ее называют функцией случайных аргументов (ФСА). Если все аргументы данной функции подчинены некоторому определенному закону распределения и результирующее распределение ФСА оказывается подчиненным тому же закону, то он называется устойчивым для данной ФСА.

Если для произвольных законов распределения аргументов при некоторых условиях (например, при $n \to \infty$) для данной ФСА можно указать

закон, к которому асимптотически стремится истинное распределение ФСА, то он называется ультраустойчивым для этой ФСА.

Рассмотрим систему, состоящую из n элементов. Случайное время до отказа i-го элемента обозначим через T_i. Время безотказной работы системы T_Σ также является случайной величиной от T_i. Исходя из этого

$$T_\Sigma=\varphi(T_1, T_2, \ldots, T_n)$$

Вид функции φ определяется структурой системы, а также принятыми определениями ее состояний работоспособности и отказа. Методика применения ФСА для определения надежности системы состоит из следующих шагов:

- построение ФСА модели рассматриваемой системы на основе имеющегося описания системы и взаимодействия ее элементов);

- запись на основе ФСА выражения для закона распределения времени безотказной работы системы T_Σ и его числовых характеристик.

Рассмотрим ФСА, типичные для задач исследования безотказности ВС, ВТ и АСУ.

1) Неизбыточная структура, все элементы которой соединены последовательно. ФСА определяется как «минимум» от времени безотказной работы элементов системы:

$$T_\Sigma=min_n(T_i), \quad i=1,2, \ldots,n$$

2) Структура с параллельным соединением элементов или нагруженный (горячий) резерв. Время безотказной работы этой модели может быть определено как ФСА «максимум» от времени безотказной работы элементов системы:

$$T_\Sigma=max_n(T_i), \quad i=1,2, \ldots,n$$

3) Структура с резервированием замещением («холодный» резерв). ФСА определяется как сумма времени безотказной работы элементов системы:

$$T_\Sigma=\sum_n T_i, \quad i=1,2, \ldots,n$$

4) Мажоритарная структура, при которой отказ системы может быть при отказе более половины ее элементов:

$$T_\Sigma=maj_n(T_i), \quad i=1,2, \ldots,n$$

5) Общий класс избыточных структур («d- безотказные структуры»). К ним относится любая система, отказ которой наступает при отказе любых d из n ее элементов. Т.е. время безотказной работы d-безотказной системы равно времени безотказной работы элемента, отказавшего (d+1)-м по счету:

$$T_\Sigma=min_n^d(T_i), \quad i=1,2, \ldots,n$$

При d=0 d-безотказная структура сводится к неизбыточной, при d=n-1- к структуре с параллельным соединением элементов, при d=(n+1)/2 - к мажоритарной. ФАС, определяющую время безотказной работы d-безотказной структуры называют ФСА «минимум d ранга».

6) ФСА «опережение» определяет время безотказной работы двухэлементной системы, у которой отказ определен как событие, состоящее в

том, что первый элемент откажет раньше второго:

$$T_{\Sigma}=on(T_1, T_2)$$

7) ФСА «отставание» определяет время безотказной работы двухэлементной системы, которая отказывает в случае, если первый элемент откажет позже второго:

$$T_{\Sigma}=om(T_1, T_2)$$

8) Другие ФСА, например, «произведение», «частное».

Метод применения ФСА заключается в следующем. Числовая характеристика ФСА n одинаковых и независимых аргументов может быть представлена в виде произведения:

$$A = h(k,n)a,$$

где a – числовая характеристика аргумента, $h(k,n)$ - функция пересчета, вид которой определяется типом ФСА и видом числовой характеристики, k-параметр формы распределения Вейбулла, представляющий собой функцию коэффициента вариации υ, k=k(υ):

$$\upsilon = \frac{\sqrt{\Gamma(1+\frac{2}{\mathrm{k}}) - \Gamma^2(1+\frac{1}{k})}}{\Gamma(1+\frac{1}{k})},$$

Функции пересчета определяются для математического ожидания, дисперсии и коэффициента вариации. Далее определяется числовая характеристика ФСА $A = h(\chi,n)a$, где $h(\chi,n)$ - функция пересчета (находится из таблицы на пересечении строки, соответствующей виду ФСА, и столбца искомой числовой характеристики); χ - находится по таблице по заданному значению υ аргумента; n- число аргументов.

Для оценки среднего времени безотказной работы системы выбираем ФСА, соответствующую типовой структуре, определяем значение функции пересчета математического ожидания η для найденной ФСА. Если распределение Т элементов не относится к вейбулловскому, и если не задан коэффициент вариации для элемента, то вычисляем этот коэффициент по формуле:

$$\upsilon = \frac{\sigma}{\mathrm{T_i}},$$

где σ - среднеквадратическое отклонение времени безотказной работы элемента ВС. Пользуясь таблицей, по вычисленному значению υ находим величину χ. Если элементы имеют вейбулловское распределение Т, то значению параметра распределения k по таблице определяем величину χ. В выражение для функции пересчета подставляем значение χ и числа элементов системы n и получаем численное значение η. Определяем среднее время безотказной работы ВС:

$$\bar{T}_{\Sigma} = \eta\, \bar{T}$$

1.6. Методы повышения надежности на этапах жизненного цикла системы

Качество организации и обслуживания авиаперевозок пассажиров и грузов в гражданской авиации зависит от надежности функционирования на всех стадиях жизненного цикла вычислительных систем и сетей, используемых для обеспечения безопасности управления воздушным движением и систем наземного обслуживания. Жизненный цикл системы (system life cycle) - это деятельность всех обеспечивающих систем, ведущих целевую систему от её замысла до вывода из эксплуатации. Надежность работы объектов отрасли гражданской авиации (от воздушных судов до вычислительных систем и сетей) характеризуется безотказностью, продолжительностью или объемом выполненной работы (наработка) и работоспособностью. Сохранение работоспособного состояния объекта характеризуется резервированием (применение дополнительных средств для повышения надежности объекта), ремонтопригодностью (возможность проведения технического обслуживания и ремонта) и сохранением значения показателей безотказности, долговечности и ремонтопригодности.

Предупреждение отказов и неисправностей, а также обеспечение надежного действия объектов должны быть обеспечены техническими решениями на этапах проектирования и производства, а также качественным содержанием объектов в эксплуатации.

При проектировании вычислительной системы изменение ее структуры в целях улучшения характеристик или обновления, появляется опасность снижения готовности, поэтому необходимо рассчитывать коэффициент готовности проекта системы на каждом шаге итерации проектирования. Характеристики функционирования сети в этом случае предлагается представить как функцию структурно-функциональной организации и коэффициента оперативной готовности, не ниже заданного:

$$H(t)=f(SF,\ Rs \geq Rz,\ C) \qquad (4)$$

где *SF* - структурно-функциональные параметры вычислительной сети, *Rs* , *Rz*- коэффициенты оперативной готовности. Условие $Rs \geq Rz$ *ограничивает* вероятность безотказной работы ВС.

Помимо обеспечения надежности аппаратных средств необходимо давать оценку и прогнозировать надежность программных средств [Заренин, РОик], составляющих значительную часть современных систем различного назначения.

1.6.1. Надежность программного обеспечения

Термином *программа* будем обозначать программное изделие, предназначенное для выполнения одной четко определенной функции. Например, программа бронирования авиабилетов, программа обработки стыковочных рейсов, программа работы с тарифной системой, программа формирования полетного плана и т.д. При этом объем и сложность этой функции (и, в частности, возможность расчленения ее на ряд более простых

функций) не играет роли. Аналог – техническое устройство или объект определенного функционального назначения [5].

Под термином *программные средства* (ПС) условимся понимать программное изделие произвольного объема и сложности и произвольного функционального назначения, предназначаемые для использования или используемые в составе других программных изделий более широкого функционального назначения. Аналог – технические средства.

Программное средство минимальной допустимой сложности, не разделяемые на более простые составные части, называется п*рограммным модулем*. Аналог – деталь, элемент.

Программное изделие, рассматриваемое как совокупность взаимосвязанных (непосредственно или косвенно) программных средств, будем называть *программной системой*. Взаимосвязь программных средств может состоять в использовании общей базы данных или одних и тех же вычислительных ресурсов, в использовании одними программными средствам и результатов работы других средств и т.п. Аналог – техническая система, комплекс технических средств.

Отдельная программа или программная система, предназначенная для решения некоторой крупной технической, специальной или какой- либо иной задачи или входящая составной частью в некоторую программно- управляемую техническую или человеко-машинную систему, называется *программным обеспечением.*

Применительно к программным средствам, говоря об их надежности, чаще всего имеют в виду не переход ПС из одного состояния (работоспособного) в другое (неработоспособное), а проявление от момента создания содержащихся в нем ошибок, вызванное попаданием на те ветви ПС, в которых находятся ошибки. Это понимание надежности как раз и отличается в существенных чертах от понимания надежности, устоявшегося в теории надежности. Таким образом, явления, которые принято относить к областям надежности технических средств и надежности ПС, принципиально отличаются друг от друга.

В соответствии со стандартом ISO 9126 под *надежностью программного обеспечения* (ПО) будем понимать комплексное свойство выполнять заданные функции, сохранять свои характеристики в установленных пределах (из заданного множества, определяемого функцией этого ПС) при определенных условиях эксплуатации (заданной вычислительной среде).

Надежность программного обеспечения (или просто программы) определяется его (ее) безотказностью и восстанавливаемостью. Безотказность программы или программного обеспечения есть ее (его) свойство сохранять работоспособность при использовании в процессе обработки информации на ЭВМ.

Безотказность программного обеспечения можно оценить вероятностью его работы без отказов при определенных уровнях внешней

среды в течение заданного периода наблюдения. В данном определении под отказом программы или системы программного обеспечения понимается недопустимое отклонение характеристик процесса функционирования программы от требуемых. Определенные условия внешней среды понимаются как совокупность входных данных и состояния вычислительной системы. Заданный период наблюдений соответствует, как правило, необходимому периоду для выполнения решаемой на машине задачи.

Безотказность программного средства можно также характеризовать средним временем между возникновениями отказов в функционировании программы. При этом предполагается, что аппаратура ЭВМ находится полностью в работоспособном состоянии.

С точки зрения надежности принципиальное отличие программного обеспечения от аппаратуры состоит в том, что программы не изнашиваются и, следовательно, их выход из строя из-за поломки невозможен. Поэтому характеристики функционирования программного обеспечения зависят только от его качества, предопределяемого процессом разработки.

Безотказность программного обеспечения определяется его корректностью (правильностью) и, следовательно, целиком зависит от наличия в нем ошибок, внесенных на этапах его создания, в то время как безотказность аппаратуры определяется в основном случайными отказами, зависящими от изменений параметров аппаратуры, происходящих во время эксплуатации.

1.6.2. Понятие некорректности программы

Под *корректностью* программы понимают её соответствие некоторому эталону или совокупности формализованных эталонных правил и характеристик [5].

Наиболее полным эталоном корректности программ является *программная спецификация*. Её особенностью является задание требований поведения программы для допустимых наборов входных данных. Поэтому корректная программа может неправильно работать или даже сбиваться на недопустимых наборах входных данных.

Свойством устойчивости к недопустимым наборам входных данных обладает надежная программа - в этом заключается разница между надёжной и корректной программами.

Требования к корректности делятся в зависимости от двух типов критериев качества:

- для функциональных критериев они определяются предметной областью и функциями выполняемой программы;
- для конструктивных критериев они определяются общими для всех программ свойствами.

В зависимости от проверяемых компонентов программ различают следующие виды их корректности:

1) корректность текстов программ имеет только конструктивную составляющую; благодаря жёстким правилам языков программирования синтаксическая и семантическая корректность программ проверяется на этапе трансляции программы, и прошедшая трансляцию программа является корректной с этой точки зрения;

2) корректность программных модулей имеет и конструктивную и функциональную составляющие: конструктивная составляющая определяется правилами построения структуры программных модулей, задаваемыми в технологии и языке программирования; функциональная составляющая корректности модулей зависит от предметной области и функциональных спецификаций программы.

Функциональная составляющая корректности может проверяться в различных условиях:

- детерминированная - для фиксированных наборов входных данных должны быть получены конкретные значения результатов;

- стохастическая - входные данные задаются случайными величинами с известными законами распределения и результаты также должны быть случайными величинами с требуемыми законами распределения и заданными корреляционными связями между входными и выходными данными;

- динамическая - характерна для систем реального времени и определяется согласованием во времени порядка поступления входных данных и порядка выдачи результатов выполнения программы.

3) корректность данных имеет конструктивную и функциональную составляющие: структурная корректность данных относится к конструктивной составляющей и предполагает правильность построения структурированных данных в программе: массивов, стеков, очередей и т.п.; функциональная корректность данных определяется диапазонами изменения их значений и соответствием типов полей структур типам значений данных.

4) корректность комплексов программ также имеет конструктивную и функциональную составляющие: конструктивная составляющая определяется корректностью структуры межмодульных связей по управлению и данным, определяемых в интерфейсных требованиях к программе; функциональной корректность комплекса программ определяется так же, как и функциональная корректность модулей.

1.6.3. Признаки отказов программного обеспечения

Основными причинами, непосредственно вызывающими нарушения нормального функционирования программы, являются:

- ошибки, скрытые в самой программе;

- искажения входной информации, подлежащей обработке;

- неверные действия пользователя;

- неисправности аппаратуры установки, на которой реализуется вычислительный процесс.

Скрытые ошибки программы. Специфика создания сложных программных средств состоит в том, что в процессе их отладки практически невозможно обнаружить и ликвидировать все ошибки. В результате в программах остается некоторое количество скрытых ошибок. Они могут вызвать неверное функционирование программ при определенных сочетаниях входных данных. Наличие скрытых ошибок программного обеспечения является главным фактором нарушения нормальных условий его функционирования.

Можно выделить следующие основные классы ошибок в программах.

Ошибки вычислений. Ошибки данного класса содержатся в закодированных математических выражениях или в получаемых с их помощью результатах. Примерами ошибок, относящихся к данному классу, являются неверное преобразование типов переменных, неверный знак операции, ошибка в выражении индекса, переполнение или потеря значимости при вычислениях.

Логические ошибки являются причиной искажения алгоритма решения задачи. Такого рода ошибки возникают в связи с неверной передачей управления, неверным заданием диапазона изменения параметра цикла, неверным условием и т. д.

Ошибки ввода-вывода, связанные с такими действиями, как управление вводом-выводом, формирование выходных записей, определение размеров записей и т. д. Примерами ошибок ввода-вывода являются неправильная форма ввода (вывода), ошибка в задании числа формируемых строк (страниц) при печати, отсутствие признака конца файла и т. д.

Ошибки манипулирования данными. Примерами таких ошибок являются неверно определенное число элементов данных; неверные начальные значения, присвоенные данным; неверно указанные длина операнда, имя переменной и т. д.

Ошибки совместимости связаны с отсутствием совместимости с операционной системой или другими прикладными программами, используемыми в данной программе.

Ошибки сопряжений вызывают неверное взаимодействие программы с другими программами (подпрограммами), с системными программами, устройствами ЭВМ, входными данными и т. д. В качестве примеров ошибок сопряжения могут быть названы несовместимость аргументов и параметров подпрограммы, отсутствие в системе необходимой подпрограммы, нарушения синхронизации при асинхронном выполнении программ и т. д.

Искажение информации, подлежащей обработке, вызывает нарушение функционирования программного обеспечения, когда входные данные не попадают в область допустимых значений переменных программы. В этом случае между исходной информацией и характеристиками программы возникает несоответствие.

Причинами искажения вводимой информации могут быть, например, следующие:

- искажение данных на первичных носителях информации;
- сбои и отказы в аппаратуре ввода данных с первичных носителей информации;
- шумы и сбои в каналах связи при передаче сообщений по линиям связи;
- сбои и отказы в аппаратуре передачи или приема информации;
- потери или искажения сообщений в буферных накопителях вычислительной системы;
- ошибки в документации, используемой для подготовки вводимых данных;
- ошибки пользователей при подготовке исходной информации.

Неверные действия пользователя, приводящие к отказу в процессе функционирования ПО, связаны, прежде всего, с неправильной интерпретацией сообщений, с неправильными действиями пользователя в процессе диалога с ЭВМ и т. д.

Отказы ПО, обусловленные ошибками пользователя, нарываются *ошибками использования.* Часто эти ошибки являются следствием некачественной программной документации (неверное описание возможностей программы, режимов работы, форматов входной и выходной информации, диагностических сообщений и т. д.).

Неисправность аппаратуры. Неисправности, возникающие при работе аппаратуры, используемой для реализации вычислительного процесса, оказывают определенное влияние на характеристику надежности ПО. Появление отказа или сбоя в работе аппаратуры приводит к нарушению нормального хода вычислительного процесса и во многих случаях - к искажению данных и текстов программ в основной и внешней памяти.

Следствием появления ошибки в программе является ее отказ, заключающийся в отклонении от выполнения программой заданных функций. В зависимости от степени серьезности последствий ошибок (отказов) в программе эти отклонения можно разделить следующим образом:

- полное прекращение выполнения функций на длительное или неопределенное время;
- кратковременное нарушение хода вычислительного процесса.

Степень серьезности последствий ошибок в программе может быть оценена соотношением между длительностью восстановительных работ, которые необходимо произвести после отказа в программе, и динамическими характеристиками объектов, использующих результаты работы программных средств. К таким характеристикам объектов относятся, например, инерционность объектов, выступающих в качестве источников и потребителей информации; заданная частота решения задач обработки

информации; заданное время реакции вычислительной системы на запросы пользователей и др.

Наиболее типичными симптомами появления ошибок в программе являются:

- преждевременное (аварийное) окончание выполнения программы;
- недопустимое увеличение времени выполнения программы;
- зацикливание ЭВМ на выполнении некоторой последовательности команд одной из программ;
- полная потеря или значительное искажение накопленных данных, необходимых для успешного выполнения решаемых задач;
- нарушение последовательности вызова отдельных программ, в результате чего происходит пропуск необходимых программ либо непредусмотренное обращение к программам;
- искажение отдельных элементов данных (входных, выходных, промежуточных) в результате обработки искаженной исходной информации.

1.6. 4. Модели надежности программного обеспечения

Надежность ПО имеет две стороны: оценка и обеспечение. Рассмотрим, как выполняется оценка надежности ПО. Она решается применением моделей, которые можно разделить на две группы: аналитические и эмпирические. Аналитические модели в свою очередь делятся на две группы: статические и динамические.

Статические модели не используют такого параметра как время, а связывают количества ошибок с числом тестовых прогонов или используют зависимость количества ошибок от характеристики входных данных. Первый тип моделей объединяют в группу с названием по области ошибок», а вторую - «по области данных». К ним можно отнести модель Миллса, модель Липова, простая интуитивная модель, модель Коркорэна, модель последовательности испытаний Бернулли, модель Нельсона [11].

Динамические модели - используют прогнозные модели, к которым можно отнести модель Шумана, модель Ла Падула, модель Джелинского-Моранды, модель Шика-Вольвертона, модель Муса, модель переходных вероятностей.

Существует другая классификация, выделяющая три группы моделей надежности ПО:

- *прогнозные* - класс моделей формируется на этапе проектирования, и они позволяют рассчитать характеристики надежности программного средства до начала его отладки. Одна из метрик Холстеда (уравнение числа ошибок) принадлежит к этой группе. Модель Холстеда дает прогнозирование количества ошибок в программе в зависимости от ее объема и таких данных, как число операций (n_1) и операндов (n_2), а также их общее число (N_1, N_2);
- *оценочные* - модели строятся на основе анализа результатов тестирования программ. Они позволяют на основе полученных значений характеристик надежности принимать решение о необходимости продолжать

процедуру тестирования программного средства. К этой группе относятся такие модели: Джелински - Моранды, Миллса и простая интуитивная или эвристическая модель двух независимых групп тестирования Руднера;

- *измерительные* - строятся на этапе испытания программного обеспечения, его сопровождения и эксплуатации (если это предусмотрено соответствующей документаций). Примеры - модели Нельсона и Муса.

Как и модель Джелински-Моранды, модель надёжности Миллса относится к классу оценочных. По другой классификации эта модель относится к классу аналитических статических моделей, и ее применение предполагает внесение искусственных ошибок перед началом тестирования. Он фиксируются в специальном протоколе искусственных ошибок. При проведении тестирования могут быть найдены как искусственные, так и естественные ошибки, которые и должны быть выявлены в результате этой процедуры. Логичным предположением применения модели Миллса является тот факт, что как естественные, так и искусственные имеют равную вероятность быть обнаруженными в процессе тестирования. По этой методике, тестируя программу в течение некоторого времени, собирают данные об экспериментах. Однако методика Милса предполагает вычисление и еще одной полезной величины - мерой доверия к модели. Она рассчитывается по формуле (5):

$$C = \begin{Bmatrix} 1, & \text{если n} > \text{K} \\ \dfrac{\text{S}}{\text{S}+\text{K}+1} & \text{если n} =< \text{K} \end{Bmatrix} \quad (5),$$

где N - первоначальное число ошибок в программе, S - количество искусственно внесенных ошибок, n - число найденных собственных ошибок, *K* - величина предполагаемого количества ошибок в программе. Величина *C* оценивает вероятность правильного значения N.

Модифицированную модель Миллса предложил Липов, включив в нее вероятность обнаружения ошибки при использовании различного числа тестов. В ней, как и в модели Миллса считается, что собственные и искусственные ошибки имеют равную вероятность быть найденными. В методике Липова предложена следующая формула (6):

$$Q(n,V) = \left(\frac{m}{n+V}\right) * q^{n+V} * (1-q)^{n-n-V} * \frac{\dfrac{N}{n} * \dfrac{S}{V}}{\dfrac{N+S}{n+V}} \quad (6)$$

где *N*- первоначальное число ошибок в программе, *m* - количество тестов, *S* - количество искусственно внесенных ошибок, *n* - число найденных собственных ошибок, *V* - число обнаруженных к моменту оценки искусственных ошибок. Применение рассматриваемой модели Липова требует выполнения следующие условия: *N>n>0; S>V>0; m>n+ V>0.* Достоинством модель Липова является то, что она позволяет оценить вероятность обнаружения некоторого количества ошибок к моменту оценки.

Простая интуитивная или эвристическая модель двух независимых групп тестирования Руднера относится к типу «разметка ошибок» (искусственное внесение в программное обеспечение известных ошибок). Она так же относится к группе оценочных, но в этой модели исключается основной недостаток модели Миллса - предположение о том, собственные ошибки, имеющиеся в программе и искусственно вносимые при тестировании имеют одинаковую вероятность. Для этой модели используются данные в предположении, что тестирование осуществляется двумя независимыми группами.

Модель надежности программного обеспечения Джелински-Моранды основана на методе максимального правдоподобия. Она относиться к классу оценочных моделей и применяется на этапе тестирования. В ее основе положены следующие предпосылки: экспоненциальная зависимость между плотностью вероятности интервалов времени между появлением ошибок; интенсивность ошибок линейно зависит от количества оставшихся ошибок (на любом случайном интервале); после каждого появления устраняют ошибку и не вносят новую; каждый тест находит только одну ошибку. Еще одно условие применимости модели Джелински-Моранды – это соответствие результатов тестирования допущению об уменьшении интенсивности ошибок после устранения очередной ошибки, то есть количество тестов (интервал времени для обнаружения каждой последующей ошибки) увеличивается.

Модель Шумана основана на следующих допущениях: общее число команд в программе на машинном языке постоянно; в начале компоновочных испытаний число ошибок равно некоторой постоянной величине, и по мере исправления ошибок их становится меньше. В ходе испытаний программы новые ошибки не вносятся; ошибки изначально различимы, по суммарному числу исправленных ошибок можно судить об оставшихся; интенсивность отказов программы пропорциональна числу остаточных ошибок. Предполагается, что до начала тестирования (т.е. в момент $\tau=0$) имеется M ошибок. В течение времени тестирования τ обнаруживается $\varepsilon_1(\tau)$ ошибок в расчете на одну команду в машинном языке. Тогда удельное число ошибок на одну машинную команду, оставшихся в системе после времени тестирования τ, равно:

$$\varepsilon_2(\tau) = \frac{M}{I} - \varepsilon_1(\tau) \quad (7)$$

где I - общее число машинных команд, которое предполагается постоянным в рамках этапа тестирования.

Предполагается, что значение функции количества ошибок $Z(t)$ пропорционально числу ошибок, оставшихся в программе после израсходованного на тестирование времени τ:

$$Z(t) = C * \varepsilon_2(\tau) \quad (8),$$

где C - некоторая постоянная, t - время работы программы без отказов. Тогда, если время работы программы без отказа t отсчитывается от точки t = 0, а τ

остается фиксированным, функция надежности, или вероятность безотказной работы на интервале от 0 до t, равна:

$$t_{cp} = \frac{1}{C * (\frac{M}{I} - \varepsilon_1(\tau)} \quad (9)$$

В процессе тестирования собирается информация о времени и количестве ошибок на каждом прогоне, т.е. общее время тестирования τ складывается из времени каждого прогона: $\tau = \tau_1 + \tau_2 + \tau_3 + \ldots + \tau_n$. Предполагая, что интенсивность появления ошибок постоянна и равна λ, можно вычислить ее как число ошибок в единицу времени:

$$\lambda = \frac{\sum_{i=1}^{n} A_i}{\tau} \quad (10),$$

где A_i - количество ошибок на i - ом прогоне.

1.6.5. Методы повышения надежности программного обеспечения

Один из способов повышения надежности ПО, заключается в определении интенсивности отказов изделий, длительности функционирования, устранении дефектов и совершенствовании ПО и его составных элементов, отличается тем, что дополнительно определяют скорости изменения интенсивности отказов и длительности функционирования от текущего времени i-го этапа цикла с количеством (n), удовлетворяющим условию n≥1, совершенствования изделия и его элементов, параметрическую взаимосвязь между ними согласно функционала.

Указанный способ обладает существенным недостатком, а именно эволюционным характером совершенствования изделия, определяемым естественными причинами, особенно в ракетно-космической технике (РКТ), такими как уникальность изделий, малая серия или единичные экземпляры, продолжительность создания (отработки) и их высокая стоимость. Все это ограничивает возможности принятия решений по использованию ряда мероприятий, направленных на совершенствование изделий, в том числе и инновационных решений.

В [12] предложен способ повышения надежности ПС, включающий определение интенсивности отказов (δ) ПО, длительности функционирования (Te), устранение дефектов и совершенствование ПО и его составных элементов. Данное решение отличается тем, что дополнительно определяют скорости изменения интенсивности отказов (Vδ) и длительности функционирования (Vc) от текущего времени (T) i-го этапа жизненного цикла с количеством (n), удовлетворяющим условию n≥1, совершенствования изделия и его элементов, параметрическую взаимосвязь между ними согласно функционала (f):

$$\delta = f(\delta o, Tco, Tcи, Tc, V\delta, Vc),$$

где (δo, Tco) - начальное значение интенсивности отказов, длительности функционирования,

$$V\delta = d\delta / dT, Vc = dTc / dT$$

- их скорости, (Тси) - координаты изменения скорости Vδ на i-м этапе цикла и/или изменение ее фазы в цикле, и функциональной связи между ними, затем с момента не далее Тси на этапе (i+1) цикла дорабатывает ПО и его элементы с обеспечением заданного не менее двукратного снижения интенсивности отказов на этом этапе продолжительностью ΔТс не более ширины спектра со средним уровнем отказов, определяемым функционалом (f), путем повышения отказоустойчивости, отказавших на i-м этапе и ранее элементов не ниже двукратного значения, где к - число элементов, при этом продолжение i+1 этапа цикла осуществляют по алгоритму i-го этапа того же цикла. При этом параметрическую взаимосвязь между параметрами в цикле совершенствования изделия, его составных элементов формируют согласно выражению

$$\delta = | a * V\delta + Vc * \delta o * Tco / (Tc - Tco) |,$$

где а* - коэффициент пропорциональности - нормировочный интервала времени множитель, при линейной зависимости от времени (Т) накопленной интенсивности отказов и длительности функционирования, при этом продолжительность i-го этапа цикла доработки не превосходит величины (Vc*δo*Tco/a*Vδ), а координаты изменения фазы циклов определяют по выражению (Тсin=Tco*(1+nδo Vc/a*Vδ), где n - номер цикла.

Контрольные вопросы

1. Назовите виды технического состояния вычислительной системы.
2. Что понимается под термином надежность системы?
3. Назовите основные эксплуатационные характеристики ЭВМ, систем и сетей.
4. Приведите классификацию методов аналитической оценки надежности.
5. Поясните, как выполняется оценка параметров надежности технических систем с использованием графов.
6. Приведите классификацию систем массового обслуживания.
7. Приведите обобщенную структурную схему СМО.
8. В чем заключается метод декомпозиции сложных технических систем?
9. Назовите типичные задачи исследования безотказности ВС, ВТ и АСУ с использованием ФСА.
10. Чем отличается надежность технических и программных средств?
11. Перечислите и поясните виды корректности программных компонент.
12. Назовите типичные симптомы появления ошибок в программе.
13. Что понимается под отказом программных средств?
14. Приведите классификации моделей программного обеспечения.
15. В чем отличие статических и динамических моделей надежности?
16. Какая модель надежности основана на методе максимального правдоподобия?
18. Перечислите условия применения модели Джелински-Моранды.

РАЗДЕЛ 2. КОНТРОЛЬ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ И СЕТЯХ

2.1. Общая характеристика систем контроля вычислительных систем и сетей

Надежность и живучесть вычислительных систем тесно связана с контролем исправности аппаратуры. Контроль функционирования ЭВМ предполагает получение исчерпывающей информации об их состоянии и может быть как программным, так и аппаратным.

Программный контроль предполагает включение в исходные коды программного обеспечения дополнительных функций, имеющих математические или логические связи с основными программами. Сравнение результатов работы функций программного контроля с результатами работы всех задач вычислительной системы в целом или отдельных этапов решения задач позволяет с определенной вероятностью судить об исправности аппаратуры и достоверности результата работы.

Аппаратный контроль предполагает введение в состав ЭВМ дополнительных элементов, предназначенных для обнаружения ошибок в результатах каждой операции проверяемой этими элементами. Аппаратный контроль приводит к значительному увеличению объема аппаратуры и потребляемой мощности. В то же время резко сокращается по сравнению с программным контролем время проверки исправности аппаратуры. Это объясняется тем, что операции проверки аппаратного контроля возможно полностью совместить по времени с операциями основного программного обеспечения.

На практике часто используют смешенные методы обнаружения и исправления ошибок. При этом аппаратный контроль служит для обнаружения неисправности, а задача исправления этих ошибок ложится на специальное программное обеспечение. Такой метод обладает достоинствами как аппаратного, так и программного контроля

2.2. Основные методы контроля ЭВМ

Особенность технологии обработки данных в том, что создается множество дефектов обработки, которые в конечном итоге снижают уровень качества работы ВС. Случаи искажения информации, идентифицируемые как дефекты обработки, носят вероятностный характер. Так, например, 0,4 % дефектов возникают по причине неисправности технических устройств, 21,6 % ошибок обусловлены недостатками проекта ИС, оставшийся объем - 78 % ошибок обусловлено человеческим фактором.

Одним из эффективных путей улучшения качества обработки информации является разработка и реализация методов обеспечения достоверности, целостности и конфиденциальности информации. Особую значимость проблема контроля данных приобретает в настоящее время при решении задачи поддержания целостности информации на всех фазах информационного процесса (рис.3).

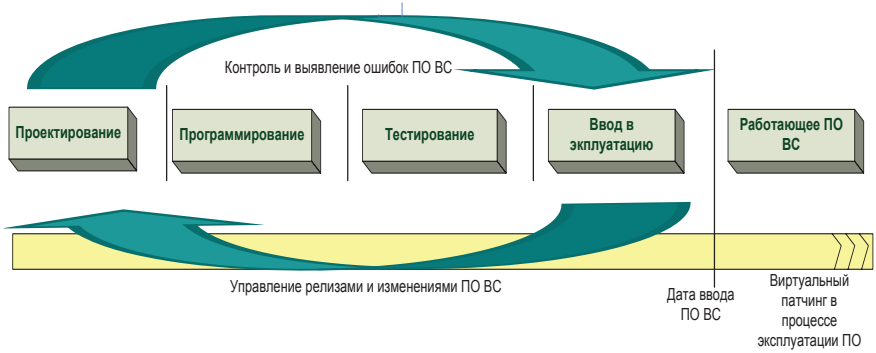

Рисунок 3- Контроль на этапах жизненного цикла

Эффективное направление - применение системы контроля обработки данных. Значительный эффект методы контроля дают в системах обработки информации числового содержания - учетных, отчетных, статистических, плановых, фактографических, параметрических и др., где искажение даже одной цифры может иметь в некоторых случаях серьезные последствия в принятии решений (например в системах формирования полетного плана).

Задача обеспечения требуемого уровня достоверности вызывает необходимость применения процедур контроля на всех основных этапах технологического процесса обработки информации. Особому контролю подвергается достоверность выходных (производных) документов, перед выдачей их пользователю. Корректировка ошибок требует привлечения довольно значительных дополнительных трудовых, материальных, финансовых и временных ресурсов. Иногда искажения в документах вызывают необходимость повторной обработки документов на ЭВМ. Для устранения подобных случаев особое внимание обращается на обнаружение и исправление ошибок на этапах сбора информации. В связи с этим большую значимость приобретает программный контроль достоверности на этапе ввода данных в ЭВМ.

Достоверность и полнота информации в ВС обеспечивается целым комплексом методов защиты: аппаратных, программных, организационных, и др. По уровню применения технических средств методы контроля достоверности информации можно разделить на следующие основные категории:

- ручной, или визуальный способ, который заключается в проверке правильности данных без применения каких-либо технических средств;

- механизированный способ состоит в применении вспомогательных технических устройств для подсчета контрольных сумм;

- автоматизированный метод контроля, заключающийся в диагностике правильности данных посредством соответствующих программных модулей;

- автоматический метод, осуществляющий программное выявление ошибочных данных, определяющий истинное значение и выполняющий замену ошибочного значения на истинное в памяти ЭВМ.

На практике обычно применяются все указанные методы. Степень их применения зависит от класса и масштаба ВС.

В значительной части ВС ввод информации в ЭВМ производится в форме документов. С целью реализации контроля достоверности входной информации разрабатываются специальные прикладные программы. Эти диагностические программы ориентированы на контроль формальных и содержательных параметров входных первичных документов. При обнаружении ошибок они выдают сообщения оператору об адресе и модификации ошибки.

Анализ работ [7,8,10] по контролю достоверности данных показывает, что имеющиеся методы и программы контроля достоверности и полноты информации ориентированы в основном на обнаружение ошибок, их идентификацию. Исправление ошибок, восстановление достоверности данных выполняется только как при непосредственном участии человека, так и автоматически.

2.3. Особенности аппаратных и программных систем контроля вычислительных систем, используемых в гражданской авиации

Рассмотрим особенности аппаратных и программных систем контроля вычислительных систем, на примере системы управления самолетом. Сегодня воздушное судно можно смело назвать «летающей» АСУТП. Современные решения для контроля и мониторинга системы управления самолетом (Flight Control System) представляют сложный программно-аппаратный комплекс, порождающий схожие проблемы с универсальным ПО, но, тем не менее, для ПО авиации существуют нюансы и типичные решения [13].

Для разработки подобных систем утверждён особый, тщательно задокументированный и регламентированный технический процесс разработки требований, создания аппаратной и программной части, выполнения и отладки системы, а так же её тестирования и составления сертификационной документации. Данный процесс постоянно модифицируется и совершенствуется с учетом новых достижений в области контроля и диагностики. Для разработки критически важных систем необходимо обеспечить минимальную возможность ошибки (для самого высокого уровня в авионике установлена вероятность отказа 10^{-9}), а так же минимизировать расходы на разработку и исправления кода.

Для контроля каждого этапа и для последующей возможности сертификации процесс разбит на различные уровни, каждому из которых соответствует свой документ, для которого создаётся в последствие документ, контролирующий его (отчёт). В итоге каждый этап разработки, все ошибки и исправления классифицируются и документируются. Повторяя каждый раз итерации разработки вероятность ошибки снижается. Данные документы так

же создаются на основе внутренних стандартов компании и требований, предъявленных заказчиком. Поэтому первая часть - это анализ требований заказчика и определение базовой функциональности системы, на основе которой создаётся общая концепция и схема системы, включая технические подробности используемого оборудования, т.е. создания первоначальных спецификаций системы (Equipment Specification) и требований (System Requirements). Далее утверждается план, по которому будет проходить разработка ПО (Software Development Plan) и его сертификация (Qualification Plan - plan for Software Aspects of Certification). Параллельным процессом является разработка аппаратной части, т.к. разработка ПО авионики очень тесно связана с аппаратной частью (в большинстве ПО является сильно зависимым от компоновки систем, несмотря на переносимый и встраиваемый код).

В основе проектирования лежат основополагающие принципы, которые присутствуют во всем процессе разработки, и главный из них - это «различие» (dissimilarity), которое определяет то, что каждая часть из систем управления должна быть реализована разными группами людей на разных аппаратных средтвах с использованием разных программных средств (в том числе средств разработки, языков программирования). Таким образом, система разделяется на программно- и аппаратно- независимые части, а процесс разработки контролируется разными группами людей для разных задач и на разных уровнях соответственно сообразно вышестоящим требованиям и плану. Результатом первичного проектирования является модель системы, на основе которой создаются документы, регламентирующие, какие аппаратные средства должны быть использованы и какие связи они должны иметь между собой.

Для взаимодействия с аппаратной частью используются драйверы и слой взаимодействия (framework layer), на основе которых должно быть построено ПО. Для высоконагруженных и отказоустойчивых ВС реалиями являются как пиковые загрузки микросхем, 90-100% загруженность шин передачи данных, синхронизация каналов, устройств и планирование загрузки фрейма в зависимости от входных параметров с контролем ошибок (в т.ч. разноуровневым мониторингом с подтверждением статичных и осцилляционных ошибок), так и ограничение размера ПО и объектов данных в объемы порядка 64-128кбайт. Разработчик ПО авионики оперирует необходимыми требованиями и архитектурой, которую он должен создать на основе требований; стандарта, определяющего общий стиль и подход к созданию архитектуры; стандарта программирования, определяющего, что разрешено писать в коде и каким стилем; задокументированных и разделённых на различные Baseline и iteration package внутри них (high-level specificaction).
По завершении каждого из Baseline проводится формирование документа SDD (Software Description Document), в котором содержится информация о дизайне приложения, а так же низкоуровневых требований, предоставленных разработчиком тестировщику.

Тестирование включает в себя низкоуровневое тестирование и высокоуровневое тестирование. Низкоуровневое тестирование включает в себя несколько этапов, в том числе: тестирование на основе требований непосредственно кода, используя метрику Маккейба и Modified Condition/Decision Coverage (используя стандарт NASA MCDC). Так же тут проверяются все граничные значения, а так же реакция системы на выход из допустимых условий (robustness testing); процесс проверки проверки, на которой проверяется правильность выполнения тестов.

Высокоуровневое тестировани включает в себя процесс создания и запуска высокоуровневых тестов на симуляторе, которые могут быть автоматическими (скриптовыми), ручными (скриптовыми, с необходимостью что-то аппаратно переключить, померить мультиметром или осциллографом), Unit-тестами (в случае, когда нельзя проверить на симуляторе).

Между каждым этапом формируется «пакет поставки», включающий в себя как спецификацию использованного оборудования, версии всех устройств и документов, так и само ПО и сопутствующие документы. Далее начинается аудит и проверка всей работы, на основе которых создаются последние три документа: First Delivery Review (FDR) - документ о поставке пакета, First Flight Review (FFR) - отчёт о первом полёте, Software Certification Review (SCR) - решение сертификационной комиссии.

В основу классификации аппаратуры контроля и диагностики положены следующие признаки [9]:

- способ управления процессом контроля и диагностики;
- вид связи контрольно-измерительной аппаратуры с диагностируемой ВС;
- принцип построения аппаратуры контроля и диагностики;
- вид обработки информации;
- назначение аппаратуры контроля и диагностики;
- степень универсальности аппаратуры контроля и диагностики;
- вид представления результатов диагностирования;
- порядок контроля параметров диагностируемой ВС.

По способу управления процессом диагностирования различают аппаратуру:

- автоматического контроля и диагностики;
- автоматизированного контроля и диагностики;
- ручного контроля и диагностики.

Аппаратура автоматического контроля и диагностики обеспечивает проведение контроля и диагностики ВС без непосредственного участия человека. Она является наиболее перспективной и все шире внедряется в процесс технического обслуживания ВС. Как правило, такая аппаратура является программно-управляемой.

Аппаратура автоматизированного контроля и диагностики обеспечивает проведение контроля и диагностики с частичным участием человека: часть программы диагностирования реализуется автоматически, часть вручную.

Аппаратура ручного контроля и диагностики обеспечивает проведение контроля и диагностики при непосредственном участии человека.

По виду связи диагностируемой ВС с аппаратурой контроля и диагностики последняя подразделяется на автономную и встроенную. Автономная аппаратура контроля и диагностики характеризуется тем, что ее функциональные системы конструктивно объединены и размещены отдельно от диагностируемой ВС. Встроенная аппаратура функционально размещена совместно с диагностируемой ВС.

По принципу построения различают аналоговую, дискретную и смешанную аппаратуру контроля и диагностики.

В аналоговой аппаратуре работа всех функциональных систем осуществляется непрерывными электрическими сигналами. В ней широко используются конструктивные элементы и принципы работы аналоговой моделирующей аппаратуры. Достоинством аналоговой аппаратуры является высокая точность переработки измерительной информации, представленной в виде напряжений или токов, и высокое быстродействие выполнения отдельных операций (сравнение, умножение, деление и др.).

В дискретной аппаратуре контроля и диагностики работа всех функциональных систем осуществляется в дискретном коде. При этом вся информация о контролируемых параметрах предварительно преобразуется в требуемый код. В качестве дискретного кода в этой аппаратуре используется двоичный код. Достоинствами дискретной аппаратуры являются высокое быстродействие выполнения элементарных операций, неограниченная точность переработки измерительной информации, сравнительная легкость реализации автоматического программно-управляемого контроля.

В смешанной аппаратуре работа одной части функциональных систем осуществляется в дискретном коде, а другой части – непрерывными электрическими сигналами. По виду обработки измерительной информации различают аппаратуру контроля и диагностики с дискретной и аналоговой обработками.

Дискретная обработка измерительной информации - это преобразование всей измерительной информации в требуемый код, выполнение логических и вычислительных операций по формированию оценок контролируемых параметров и регистрация результатов контроля и диагностики. В качестве выходных устройств при дискретной обработке используют дисплей, печатающие устройства, устройства записи на внешний носитель и др.

Аналоговая обработка измерительной информации - это преобразование всей измерительной информации в аналоговый вид, выполнение логических операций по формированию оценок контролируемых параметров, измерение параметров и регистрация результатов диагностирования. При этом в качестве

выходных устройств такой аппаратуры для представления и регистрации результата используют световые и люминофорные табло, стрелочные приборы и самопишущие регистрирующие приборы.

Смешанный вид обработки представляет собой сочетание дискретной и аналоговой обработок измерительной информации.

Аппаратура контроля и диагностики может быть предназначена для решения следующих задач диагностирования:

- контроль технического состояния, поиск неисправностей;
- прогнозирование технического состояния;
- оценка текущей работоспособности.

По степени универсальности аппаратура контроля и диагностики делится на специализированную и универсальную.

Специализированная аппаратура контроля и диагностики предназначается для диагностирования элементов ВС одного вида. Решению этой задачи подчинены внутренняя структура построения аппаратуры и программа ее работы. Достоинством такой аппаратуры является простота ее конструкции.

Универсальная аппаратура контроля и диагностики предназначается для решения нескольких задач диагностирования различных типов элементов ВС. Это достигается выбором определенной программы ее работы и структуры построения аппаратуры.

По виду представления результатов диагностирования различают аппаратуру контроля и диагностики с качественными и количественными представлениями и регистрацией результата.

Качественное представление результата диагностирования по каждому параметру и по аппаратуре в целом с оценкой типа "годен" – "негоден", "в допуске" – "не в допуске", "больше допустимого" – "в допуске" – "меньше допустимого" обычно выдается в виде визуальной индикации на табло.

Количественное представление результатов диагностирования осуществляется путем:

- визуального представления числовой информации на табло, мониторах ЭВМ и т. д.;
- вывод информации на устройства печати, на устройствах внешней памяти, и др.

По порядку контроля параметров диагностируемой ВС различают аппаратуру последовательного и параллельного контроля параметров элементов ВС.

При разработке отдельных видов аппаратуры контроля и диагностики ВС необходимо учитывать различные взаимосвязанные факторы, определяющие технические требования к ней. К числу таких факторов относятся:

- цель диагностирования и назначение аппаратуры контроля и диагностики;
- допустимое время контроля и диагностирования;

- надежность диагностируемой ВС;
- требуемая надежность аппаратуры контроля и диагностики;
- минимальное взаимное влияние ВС и аппаратуры контроля и диагностирования;
- требуемая точность измерения параметров;
- степень универсальности аппаратуры контроля и диагностики;
- форма обработки информации;
- вид представления результатов контроля и диагностирования;
- пропускная способность системы диагностирования;
- периодичность проверки аппаратуры контроля и диагностики;
- допустимое время подготовки аппаратуры контроля и диагностики к работе;
- время непрерывной работы;
- унификация конструктивного исполнения аппаратуры контроля и диагностики;
- допустимая стоимость аппаратуры контроля и диагностики;
- простота управления и эксплуатации аппаратуры контроля и диагностики, обслуживающий персонал.

Сущность указанных факторов, получение их количественных оценок и выделение взаимосвязи между собой совместно с техническими данными, характеризующими ВС, обусловливают разработку аппаратуры контроля и диагностики, наиболее полно отвечающую задачам обеспечения технического обслуживания ВС.

2.4. Методы организации контроля информативных диагностических параметров

Возможны два подхода, реализующих контроль технического состояния ВС и сетей по диагностическим параметрам [8].

Первый из них заключается в организации постоянного контроля изменений информативных параметров (трендов). Трендовая характеристика позволяет прогнозировать момент наступления катастрофических изменений технического состояния, и, следовательно, планировать и прогнозировать остаточный ресурс и планировать срок физически обоснованного ремонта. Этот способ рекомендуется для контроля технического состояния дорогостоящих или ответственных объектов, нарушение которых может привести к катастрофическим последствиям. Поэтому на этапе эксплуатации технических средств ВС и сетей большое внимание уделяется прогнозной диагностике зарождающихся дефектов элементов ЭВМ, ВС и сетей.

Для этого используют тесты: контролирующие, они определяют неисправное состояние объекта контроля и диагностирования, и диагностирующие, локализующие место расположения неисправности, для заданного множества неисправностей $S=\{s_i\}$. Множество П$к$ контролирующего теста должно обладать свойством обнаружения прогнозируемых параметров отклонения от нормы.

2.5. Оптимизация периодичности контроля

В процессе хранения и эксплуатации технических систем самого разнообразного назначения часто возникает задача (при конечном времени контроля систем) выбора периода контроля их состояния в условиях, когда каждая из систем не может быть проверена в промежутках времени между моментами контроля. Если предположить, что система не может быть использована по назначению при контроле ее технического состояния, то частый контроль снижает готовность системы к работе. Однако и достаточно редкий контроль также приводит к снижению готовности системы, так как в этом случае большое время система не может быть использована по назначению по причине ненадежности (отказ системы выявляется и устраняется только при проведении контроля). В рассматриваемой ситуации должен существовать оптимальный период контроля системы. Описанная физическая задача типична для всех ВС, а также для всех систем, недоступных частому или непрерывному контролю из-за отсутствия встроенных индикаторов, невозможности проверок со стороны оператора. Например, в системах автоматической обработки информации с записью информации в долговременное запоминающее устройство потеря информации происходит как от момента отказа до момента контроля, так и при проведении контроля. Аналогичная ситуация возникает и при работе некоторых авиационных и других систем.

Перейдем к математической постановке задачи. Предположим, что время контроля исправной системы равно V, неисправной равно U, время от момента окончания контроля до отказа системы X, а время между последовательными моментами контроля (от момента окончания предыдущего контроля до начала последующего контроля) Y. Будем вначале считать, что все эти времена являются случайными с соответствующими функциями распределения $F_v(t)$, $F_u(t)$, $F(t)$, $G(t)$.

Обозначим через Z следующую величину:

$$Z = \begin{Bmatrix} Y+V, \text{если } X > Y, \\ Y+U, \text{если } X < Y \end{Bmatrix}$$

Основное допущение будет состоять в предположении того факта, что величины $\{Z_k\}$ образуют процесс восстановления с $\mu = M[Z]$ и функцией восстановления $H(t) = M[N_t]$, где $N_t = max\ n$. Здесь n- число, при котором $S_n \le t$, $S_n = z_1 + z_2 + \ldots + z_n$ [Б]. Тогда дифференциал $dH(t)$ можно трактовать как вероятность того, что окончание контроля произойдет в момент t. Действительно, для малого отрезка ∇t имеем

$$MN_{\nabla t} = 0p_0(\nabla t) + 1p_1(\nabla t) + 2p_2(\nabla t) + \ldots \approx p_1(\nabla t) + o(\nabla t),$$

где $p_1(\nabla t)$ – вероятность того, что окончание контроля системы произойдет на малом отрезке времени (∇t).

Эффективность обнаружения отказов ВС определяется полнотой оперативного и периодичностью тестового контроля. Уменьшение интервалов периодичности контроля приводит к снижению готовности системы из-за

роста временных издержек на тестирование, но в то же время повышает ее безопасность в результате снижения вероятности функционирования системы в состояниях необнаруженных отказов. В системах с дублированием компьютерных узлов возможны режимы разделения нагрузки, когда узлы независимо выполняют распределяемый между ними поток запросов, и режим дублирования вычислений, когда каждый запрос одновременно выполняется двумя компьютерными узлами при сравнении результатов в контрольных точках. Для дублированных систем с разделением нагрузки имеется потенциальная возможность повышения эффективности контроля в результате периодического перехода в режим дублированных вычислений со сравнением результатов, что позволяет уменьшить издержки на проведение тестового контроля (дублированной системы), инициируя его только в случае несовпадения результатов дублированных вычислений. Основная задача состоит в определение оптимальных интервалов перехода в режим дублированных вычислений для обеспечения максимума вероятности готовности системы к безопасному выполнению функциональных запросов при минимизации простоев и задержек обслуживания. В [11] описана марковская модель, позволяющая определить вероятность состояний системы, в том числе готовности системы к безопасному функционированию, простоев и опасных состояний необнаруженных отказов. Данная модель позволяет анализировать влияние периодичности инициализации режима дублированных вычислений на готовность системы к безопасной работе и рассчитать оптимальную периодичность инициализация режима дублированных вычислений, при которых достигается максимум вероятности готовности системы к безопасному функционированию при минимизации простоев ВС.

Контрольные вопросы

1. Что включает контроль функционирования ЭВМ?
2. Какие задачи диагностирования выполняются аппаратурой контроля?
3. Поясните основные методы контроля ЭВМ.
4. Назовите назначение специализированной аппаратуры контроля.
5. Какие факторы необходимо учитывать при разработке отдельных видов аппаратуры контроля и диагностики ВС?
6. Какие признаки положены в основу классификации аппаратуры контроля и диагностики?
7. Назовите задачи низкоуровневого и высокоуровневое тестирования.
8. Поясните используемую систему контроля ВС на этапах разработки.
9. Назовите критерии оптимизации периодичности контроля.
10. Поясните математическую постановку задачи контроля.
11. Какая возможность имеется в дублированных систем с разделением нагрузки?

РАЗДЕЛ 3. СИСТЕМЫ ДИАГНОСТИКИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

3.1. Термины и определения

Для реализации обслуживания технических объектов по состоянию необходимы методы и средства технического диагностирования, которые дают возможность непрерывно или периодически определять действительное состояние объекта [2]. Начиная с 80-х г.г. XX столетия наблюдался эволюционный скачок в области создания систем мониторинга, диагностических систем, обеспечивающих значительное увеличение информации о текущем техническом состоянии (ТС) объектов.

Прогнозирование ТС - определение вида ТС объекта или параметров, характеризующих ТС с некоторой (иногда заданной) вероятностью на предстоящий интервал времени (ресурс) или установление интервала времени с определенной вероятностью, в течение которого сохраняется работоспособность объекта или наступает отказ.

Виды неисправностей - повреждения, нарушения функционирования и дефекты. Повреждения относятся к нарушению исправного состояния в эксплуатации. Неправильное функционирование – нарушение алгоритмов функционирования в эксплуатации и при изготовлении. Дефект относится к определению качества изготовления.

Диагноз – результат контроля ТС – установление определенной неисправности в объекте диагностирования или отнесение объекта к определенной категории (классу) технического состояния.

Диагностическая модель - формализованное описание объекта диагностирования (ОД), учитывающее возможность изменения его состояния во времени. Модели ОД необходимы для построения алгоритмов диагностирования формализованными методами, для анализа результатов на полноту обнаружения и на глубину поиска неисправностей. Диагностические модели представляются в аналитической, табличной, векторной и графической формах.

Структурные параметры - параметры, определяющие структуру объекта.

Диагностические параметры (признаки) - параметры объекта диагностирования, которые количественно или качественно характеризуют ВС объекта. Как правило, ВС определяется по совокупности диагностических параметров.

Контролепригодность (диагностируемость) - приспособленность ОД к измерению диагностических параметров (признаков) средствами диагностирования.

Средства диагностирования - аппаратура и программы, с помощью которых осуществляется диагностирование.

Сущность технического диагностирования состоит в разработке и практической реализации алгоритмов оценки параметров технических состояний объектов диагностирования без их разборки в рабочих условиях.

Структурные свойства ОД. Если под техническим состоянием (ТС) объекта диагностирования понимать *совокупность его свойств*, изменяющихся во время работы, то для поиска дефектов, прогнозирования ТС необходимо выделить те существенные свойства объекта, которые обеспечивают его устойчивое функционирование и являются в связи с этим характеристикой ТС. Кроме этого необходимо найти закономерности изменения этих свойств и тем самым определить виды ТС, в которых может находиться объект диагностирования.

Объект технического диагностирования (контроля состояния) **-** изделие и (или) его составные части, подлежащие диагностированию (контролю).

Условная вероятность необнаруженного отказа (неисправности) при диагностировании (контроле) **-** вероятность того, что неисправный (неработоспособный) объект в результате диагностирования (контроля) признается исправным (работоспособным)

Условная вероятность ложного отказа (неисправности) при диагностировании (контроле) - вероятность того, что исправный (работоспособный) объект в результате диагностирования (контроля) признается неисправным (неработоспособным)

Условная вероятность необнаруженного отказа (неисправности) в данном элементе (группе)- вероятность того, что при наличии отказа (неисправности) в результате диагностирования принимается решение об отсутствии отказа (неисправности) в данном элементе (группе)

3.2. Состав и показатели качества систем диагностики

Особенностью современных вычислительных систем и сетей является, прежде всего, широкое использование в их составе средств вычислительной техники. Как правило, это сложные системы, отличающиеся разнообразием, большим числом компонент, связей между ними, поставленными перед ними целями. Так, в современных ВС можно выделить такие основные компоненты, как аппаратура, линии связи, программное обеспечение, обслуживающий персонал, а соответственно им и типы возникающих нарушений. Частоты появления этих нарушений различные для разных структур систем, но при современном уровне технологии производства компонент весьма значительные, что существенно снижает эффективность систем такого класса и удлиняет сроки ввода в эксплуатацию.

Средствами технического диагностирования решаются такие задачи:

- обеспечение высокого уровня достоверности функционирования и защита от выдачи неправильных воздействий на управляемый объект;

- своевременное обнаружение, устранение и замена отказавших компонент ВС;

- организация рациональной эксплуатации таких систем, включая прогноз будущего состояния и обеспечение средствами восстановления работоспособности.

Каждая из указанных задач характеризуется возрастающей сложностью, обусловленной появлением новых элементов, выполненных с использованием последних достижений современных технологий. Причем существенно изменяются типы отказов и сбоев, поэтому изучение физических причин их появления - одна из основных задач для правильной ориентации разрабатываемых средств защиты.

Для количественной и качественной оценки свойств систем технического диагностирования применяют следующие характеристики и показатели качества:

1) оперативность - характеризует возможность своевременного и обоснованного выбора управляющих воздействий в процессе функционирования системы с целью учёта изменений в ситуации;

2) гибкость - определяет возможность системы перепрограммирования на различные условия и режимы работы;

3) мобильность - определяет быстроту перестройки системы с изменением состояния внешней среды;

4) живучесть - характеризует возможность временного продолжения функционирования в случае повреждения отдельных деталей и узлов.

Параметры технического состояния объекта, которые контролируются в процессе диагностирования, делятся на:

- выходные - непосредственно характеризуют работоспособность и связаны с целевым назначением объекта, а также, могут служить для его характеристик качества;

- косвенные - функционально и стохастически связаны с выходными параметрами, характеризующими возможность их оценки в процессе работы.

При диагностировании используют вектор структурных параметров объекта $R=\{r_1, r_2, r_3, \ldots, r_m\}$, где r_i , $i=1,2,3, \ldots m$ - отклонение i-го параметра технического состояния от его номинального значения.

Состояние сложного объекта диагностирования обычно оценивается множеством структурных параметров r_i, изменение которых со временем наработки приводит к отказу. На основе изучения статистических отказов механизма составляется перечень слабых узлов, лимитирующих ресурс механизма, составляется также перечень подлежащих диагностированию дефектов, обусловленных выходом того или иного структурного параметра за допустимые пределы. Нарушение работоспособности объекта диагностирования со временем наработки, называемое в теории надежности отказом, хотя и является случайным событием, обусловлено, тем не менее, вполне определенными физическими процессами, протекающими в аппаратной части ВС при ее эксплуатации и вызывающими старение и износ

элементов. Эти процессы в свою очередь зависят от ряда внешних и внутренних факторов, в том числе режимов и условий работы.

Комплекс внешних и внутренних факторов, приводит к изменениям структурных параметров. Процесс приближения технического состояния объекта диагностирования к отказам характеризуется движением случайно изменяющегося с течением времени эксплуатации ВС вектора структурных параметров R к границе рабочей области, при достижении которой объект теряет работоспособность.

Диагностическим системам приемлемы общие принципы системного анализа: принцип целеобусловленности создания системы (совокупности технических средств и обслуживающего персонала); принцип относительности (совокупность элементов системы, рассматриваемая как часть большей системы); принцип управляемости (определения возможности изменения структуры системы и иерархичности её построения); принцип модулируемости (обеспечение возможности прогнозирования состояния объекта, диагностирования или развития самой системы).

Сложные ВС отличаются также значительными сроками проектирования, за которые они успевают иногда морально устареть, большими сроками отладки и затратами материальных ресурсов. В связи с этим, особое значение имеют вопросы их автоматизированного проектирования. Автоматизированные системы проектирования должны применяться и к средствам повышения эффективности таких систем, а именно к средствам технического диагностирования. Таким образом, важное значение приобретают вопросы формализации процессов проектирования средств и систем диагностирования, что требует решения комплекса теоретических и практических вопросов. Этот комплекс вопросов и составляет проблему создания диагностического обеспечения сложных систем, к которым относятся вычислительные системы и сети.

На содержательном уровне разработка диагностического обеспечения включает в себя: анализ нарушений; оценку их влияния; разработку технических средств для их обнаружения и локализации; размещение этих средств в системе; организацию взаимодействия средств обнаружения; поиск и устранение нарушений.

Основной задачей диагностирования ВС является установление их фактического состояния для того, чтобы на основе полученной информации организовать их рациональную эксплуатацию. Организация процессов диагностирования предполагает выполнение следующих этапов:

- составление математического описания объекта;
- разработка диагностической модели (ДМ);
- анализ диагностической модели и выбор совокупности контролируемых показателей (прямых и косвенных);
- оценка достоверности выбранных показателей;
- разработка алгоритмов и программ тестирования;

- разработка средств подготовки процесса диагностирования;
- разработка средств проведения процесса диагностирования.

Каждый из указанных этапов в свою очередь состоит из ряда операций. Так, этап осуществления процесса диагностирования включает в себя следующие операции:

- выработку входных воздействий;
- подачу входных воздействий на объект диагностирования (ОД);
- управление объектом диагностирования;
- съем выходных реакций ОД;
- анализ выходных реакций ОД и принятие решений.

Этап подготовки средств диагностирования, например тестовых, состоит из следующих операций: подготовки тестов; составления словаря; оценки их качества; перенесения на носители автоматических установок контроля.

Все отмеченные этапы тесно связаны между собой. Особо важное значение имеет этап составления и анализа диагностических моделей, ибо неверное их составление приводит к созданию неэффективных средств проверки, к напрасным затратам материальных ресурсов на процесс контроля. В связи с этим желательно иметь определенные количественные характеристики каждого этапа и установления связи этих характеристик между собой. Каждый из этапов реализуется достаточно сложно.

Разработка технических средств включает в себя выбор и реализацию отдельных проверочных операций. Основным путем решения этой проблемы является широкое использование методов математического моделирования, ориентированных на машинную реализацию. Модели ВС также весьма сложны, что требует использования специальных методов их выбора, построения и анализа.

Модели систем необходимы для выбора рациональных вариантов таких систем на ранних стадиях их создания. Применительно к сложным системам трудности возникают на стадиях формального генерирования вариантов таких систем, выбора критериев и правил отсечения неудачных вариантов, что требует привлечения специальных методов и приемов, основанных, например, на принципе декомпозиции, использования агрегированных описаний, а также ряда других методов и приемов.

Для сложных систем полностью формализовать процесс создания диагностического обеспечения не удается, да и вряд ли целесообразно. Необходимо рациональное сочетание опыта проектировщика и формальных методов проектирования. В рамках такого подхода формализованные методы дают возможность частично генерировать варианты в рамках определенных принципов, задаваемых проектировщиком, формализовать процедуры отсева нерациональных вариантов. Полностью формальные методы могут использоваться для отдельных операций и проектирования отдельных устройств ВС.

Рассмотрим схемы организации процессов диагностирования таких классов систем, как ВС, сети ЭВМ, отдельные машины и т.п . В данном разделе приведены также схемы организации процессов проверок сложных технических систем (иерархические схемы), устанавливается связь процессов диагностирования и отладки в АСУ ТП. Указанные схемы являются основой для дальнейшего формализованного решения задач обеспечения контролепригодности и проверки как самих систем, так и их компонент.

3.3. Принципы организации процессов диагностирования на системном уровне

Современные ВС ГА - это прежде всего различные средства вычислительной техники, взаимодействующие с рядом специальных устройств. Представляет интерес рассмотрение процессов диагностирования для различных способов использования ЭВМ: в вычислительных системах, сетях и т. д.

3.3.1. Организация процессов диагностирования ВС, устойчивых к отказам

Существует много признаков, по которым можно классифицировать ВС: тип ЭВМ или процессоров, тип машин, способы управления, структура систем, основные режимы работы и т. д. Рассмотрим ВС реального времени, устойчивые к отказам, с разнородными машинами.

Объектом диагностирования в ВС являются аппаратура средств связи, программное и информационное обеспечение.

С помощью средств технической диагностики в ВС, устойчивых к отказам, решаются задачи:

- функционального диагностирования с целью оперативного обнаружения и классификации ошибок, возникающих в ВС;
- тестового диагностирования для поиска места неисправности с целью ее устранения;
- создания условий для предотвращения размножения отказов по системе и снижения уровня деградации системы;
- контроля не только информационных, но и физических связей.

Выполнение обычных диагностических функций в сложных ВС имеет особенности, обусловленные тем, что эти функции реализуются для связанных машин, вследствие чего возникают определенные зависимости, не имеющие места в обычных одномашинных комплексах. Так, для реконфигурации необходимо знать состояние всех машин системы и наметить новую структуру системы.

Рассмотрим организацию *тестового диагностирования* - диагностирования, при котором на объект подаются тестовые воздействия. Для этого в вычислительной системе необходимо выделить проверяющие и проверяемые подсистемы и обеспечить их связность.

Возможны следующие условные схемы организации тестового диагностирования:

- с централизованным аппаратным ядром (рис. 4,а);
- с централизованным аппаратно-программным ядром (рис. 4,б);
- с распределенным аппаратным ядром;
- с распределенным аппаратно-программным ядром;
- смешанные схемы организации тестирования (рис. 4,в).

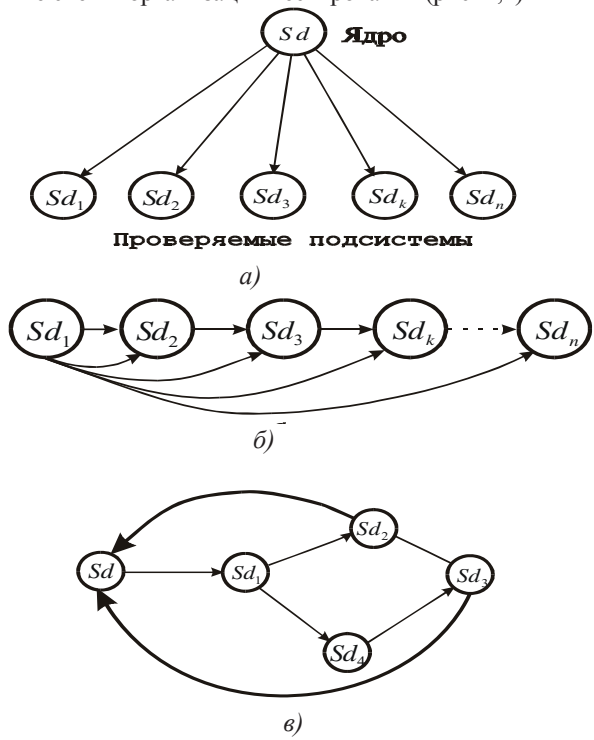

Рисунок 4 – Условные схемы организации тестового диагностирования: *а* – с централизованным аппаратным ядром; *б* – с централизованным аппаратно-программным ядром; *в* – смешанные схемы организации тестирования

Под ядром понимается совокупность в общем случае аппаратных и программных средств, обеспечивающих выполнение функций - хранение или оперативную генерацию входных воздействий, подачу их на проверяемую машину, получение реакций и сравнение их с эталонными, переход к выбору следующего набора и останов. Состав и способы реализации функций ядра могут быть различными. Ядро может быть полностью аппаратным (жестким), программным, микропрограммным, комбинированным и в свою очередь организовываться по определенным принципам. Под переменным понимается такое ядро, в котором дополнительно к заранее выделенному проверяющему устройству, включающему программные и аппаратные средства, добавляются некоторые устройства из числа проверенных.

Связи между проверяющими и проверяемыми подсистемами организуются по разным признакам. Использование того или иного вида связей требует определенных затрат и приводит к получению различного эффекта.

При централизованном принципе организации проверок в ВС назначается машина - ядро, с помощью которой поочередно проверяются все машины. Схема такой системы при полностью независимых машинах показана на рис. 4*а*, при определенном типе связей между машинами – на рис. 4*б*. Считается, что машина-ядро S_d (проверяемая машина) проверена заранее и в процессе диагностирования не отказывает. При этом число дополнительных связей $I = \alpha n$, где α - коэффициент, учитывающий тип связи (одно-, двунаправленная, $\alpha = \{1,2\}$, n - число машин.

Является естественным использование для организации проверок как существующих в системе связей, так и дополнительных, обеспечивающих различные блокировки и переключения для упрощения процесса проверки (рис. 4,*в* показано выделенными линиями).

В таких схемах организации проверок могут использоваться различные стратегии их проведения. Например, известная стратегия расширяющихся областей применяется, когда проверенные машины используются для проверки последующих с целью экономии времени контроля.

Вначале могут проверяться связи между машинами, схемы сравнения, запоминающие устройства, в которые вводятся тесты, сама тестовая информация. Затем проверяется оставшееся оборудование.

При распределенном принципе размещения машин-ядер в ВС каждое ядро может проверять другое, при этом не требуется, чтобы они обязательно были проверены и не отказали в процессе диагностирования (рис. 5).

В каждой схеме могут применяться разные стратегии проверок, например одношаговая, при которой каждая машина проверяет другую по одному разу; многошаговая, при которой возможны многократные проверки. Исходом таких проверок является синдром - признак, показывающий эффект проведения проверок (число обнаруженных одновременно неисправных машин) в зависимости от числа связей между машинами и числом проверок.

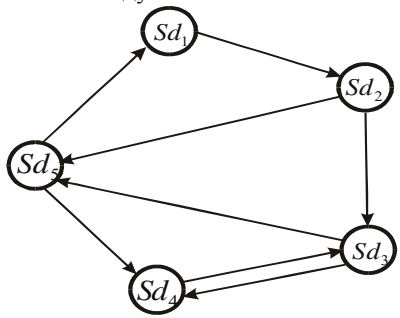

Рисунок 5 – Условные схемы организации тестового диагностирования при распределенном принципе размещения машин-ядер в ВС

Рассмотрим особенности организации *функционального диагностирования* в ВС. Необходимость применения функционального диагностирования обусловлена тем, что число сбоев в УВС значительно больше, чем в обычных ЭВМ, в которых они также составляют значительную часть (до 90 %).

К организации функционального диагностирования предъявляются те же требования, что и для отдельных машин: своевременность обнаружения ошибок и полнота охвата.

Используются следующие принципы организации проверок:

- централизованное размещение контрольных точек и устройств;
- распределенное размещение контрольных устройств. В УВС появляется необходимость защиты от ошибок, возникающих в связях между машинами.

Функциональное диагностирование также реализуется с использованием аппаратных и программных средств.

В ВС, устойчивых к отказам, средства технического диагностирования непосредственно связаны с системой реконфигурации, предназначенной для перестройки структуры системы в случае обнаружения ошибок. Система реконфигурации определяет набор машин и связи между ними с помощью информации, получаемой от средств технического диагностирования.

Система реконфигурации предъявляет следующие требования к средствам технического диагностирования:

- полный контроль ошибок, обусловленных неисправностями в аппаратуре;
- сохранность при контроле нормального функционирования системы. Система реконфигурации реализует:
- физическое переключение связей;
- действия, выполняемые операционной системой;
- действия, выполняемые оператором.

Выполнение этих действий состоит во внесении изменений в таблицы, которые хранятся в памяти и определяют конфигурацию имеющегося в системе оборудования и каналов передачи информации.

При использовании систем реконфигурации программные и аппаратные средства организуются по централизованному или распределенному принципам.

Порядок работы устройств в системах реконфигурации может быть различным. Например, в некоторых ВС имеется специальное устройство управления готовностью, которое изменяет конфигурацию системы. Принцип работы этого устройства следующий: предполагается, что произошел отказ и устройство автоматически перезапускает программу исправления ошибок всей системы в случае, когда его внутренний счетчик времени не будет сброшен исполнительной системой в течение определенного промежутка времени (1–15 с).

В другом случае операционная система проверяет наличие определенных признаков устройств, вносит изменения в системные таблицы и производит распределение ресурсов.

3.3.2. Средства организации проверок

В ВС могут применяться известные аппаратные и программные средства проведения проверок и их различные модификации. К программным средствам организации проверок относятся:

- модифицированный многократный счет сразу на нескольких машинах одновременно;
- счет на разных машинах с последующим сравнением результатов;
- измерение по времени задержки информации, проходящей через определенную машину (использование контрольного таймера);
- решение фоновых тестовых задач на машинах, свободных от выполнения основных функций;
- использование специальных команд типа «диагностика»;
- использование стохастических сигнатур для тестовых проверок;
- организация пауз в работе для исключения влияния помех.

К аппаратным средствам проведения проверок следует отнести:

- введение дополнительных процессоров, выполняющих функции диагностирования и реконфигурации;
- ведение дополнительных связей для обеспечения процесса проверок.

Следует отметить особую важность организации тестирования программного обеспечения. Различают тестирование операционных систем (ОС) и прикладных программ.

Тестирование ОС, включая тестирование всех ее элементов, проводится как с помощью программ пользователя, так и специальными методами, которые в основном рассчитаны на проверку работы системы прерывания, диспетчеризации. Поскольку ОС в вычислительных системах выполняет функции реконфигурации путем оперирования с информацией, получаемой средствами технического диагностирования, необходимо периодически контролировать выполнение этих функций либо распределять ее функции между машинами системы.

Тестирование прикладных программ также имеет важное значение, поскольку надежность ВС определяется надежностью этих программ.

В настоящее время применяют два принципа организации тестирования программ: восходящее и нисходящее. Восходящее тестирование состоит в выполнении следующих этапов: тестирование модулей, тестирование подсистем, тестирование систем, приемо-сдаточные испытания. Тестирование модулей требует базы тестовых данных и других элементов окружения, в котором будет работать модуль. При тестировании подсистем проверяются связи между модулями. На этапе тестирования систем проверяется система в целом.

При нисходящем тестировании выделяется основная программа и один–два уровня подпрограммы в качестве ядра системы, затем проверяется само ядро системы. После установления работоспособности ядра эта система наращивается путем добавления по одному из новых модулей, которые проверяются отдельно.

В рамках этих процедур применяются различные методы построения тестов программ. Одним из таких методов является построение всех возможных путей в программе и подбор входных данных, обеспечивающих проверку этих путей. Такие задачи легко решаются с помощью аппарата теории графов.

3.3.3. Организация проверок в вычислительных сетях

Под вычислительной сетью понимают совокупность вычислительных устройств, соединенных каналами связи так, что между двумя любыми вычислительными центрами (ВЦ) может быть установлена функциональная связь (либо непосредственно, либо через транзитные ВЦ). Простейший вариант ВЦ - это терминал (узел сети) и устройство связи (узел связи).

Сети различают по составу оборудования: гетерогенные, когда в их составе используются физически и логически различные машины, и гомогенные, когда применяются идентичные машины. Кроме того, сети различают по структуре связей между машинами с централизованной (звездной) топологией, децентрализованной, кольцевой и радиально-кольцевой.

Вычислительные сети подразделяют на сети с фиксированными каналами связи между ВЦ, с коммутацией каналов и сообщений (пакетов), со смешанной системой передачи данных. Работа сети производится под управлением распределенной операционной системы, составными частями которой являются протоколы и программы взаимодействия пользователя с сетью. Протоколы - это программы, которые в соответствии с заданием осуществляют управление системой передачи данных и вычислительной сетью в целом. Протоколы содержат набор соглашений о форматах и порядке следования сообщений во времени.

В отличие от обычных и распределенных вычислительных систем сети как объект диагностирования имеют ряд особенностей. Это, прежде всего появление помимо обычных отказов в машинах и их программном обеспечении специфичных ошибок и их последствий.

Особенности функционирования сетей в условиях нарушений следующие:

- неисправность или ошибка из одного узла сети может распространяться в другие узлы;
- данные и контрольная информация могут быть утеряны или искажены при передаче их через аппаратуру связи и машины;
- имеется возможность изоляции неисправного узла и резервирования его путем обхода по другим путям.

Причины отказов в вычислительных сетях для отдельных узлов такие, как и для отдельных машин - это ошибки проектирования, отказ компонентов, ошибка из-за вмешательства человека-оператора, ошибка в программном обеспечении.

Существует ряд специфичных ошибок:

- шум (в основном для коммутируемой сети с аналоговыми средствами);
- отказы компонентов связи;
- потеря синхронизации между связными процессорами, распределенными в сети, которая усугубляется большими задержками информации в линиях связи;
- неверное составление или неверное выполнение протоколов различного уровня;
- потеря данных (пакетов или сообщений), вызванная переполнением сети или нарушением алгоритмов передачи данных;
- блокировка сети, возникающая из-за противоречивых требований узлов сети один к другому, обусловленных некорректным составлением протоколов, когда никакая обработка не может производиться.

Задача тестового диагностирования в сети состоит в уточнении узла сети, в котором произошла ошибка, т. е. в получении заключения об исправности машины, ее программного обеспечения и линий связи. Эта задача решается следующими способами:

- узел обнаруживает свои нарушения самостоятельно (в этом случае применимы все методы, разработанные для изолированных систем);
- нарушения в узле обнаруживаются с помощью других узлов сети двумя путями: посылкой специальных сообщений для возбуждения собственных средств тестирования узла и посылкой специальных сообщений-тестов. Более подробная диагностика может осуществляться:
- самим узлом;
- посылкой информации для центра обслуживания системы, где хранится база данных, называемая *словарем*, с целью уточнения места неисправного блока в узле.

Неисправный узел можно устранять автоматически, исключая его из сети и заменяя исправным либо временно исключая узел и ремонтируя его с помощью обслуживающего персонала.

Достоверность функционирования сети зависит от методов оперативного обнаружения ошибок. Могут использоваться традиционные методы проверки аппаратуры и программного обеспечения, основанные на введении информационной избыточности (кодов с обнаружением и исправлением) в процессоры и линии связи. Возможны способы проверки, свойственные только сетям. Например, можно получить из некоторого узла неправильное сообщение или не получить сообщения вообще за ожидаемое время, узел может не принять переданное ему сообщение. Эти методы

реализуются благодаря использованию информационной избыточности протоколов и внешних процедур.

Один из путей определения неисправности в сети состоит в периодическом анализе состояния всех узлов и линий связи в сети и сообщении о неисправностях в контрольный центр сети.

Структуры данных в сети могут быть продуманы таким образом, чтобы вносить в них избыточность для обнаружения ошибок. Внесение избыточности осуществляется традиционными способами.

Защита управления и данных в сети может осуществляться обычными методами (дублированием, кодами с исправлением ошибок).

Защита от последствий неправильных действий в сети осуществляется такими методами:

- предупреждение процесса деградации путем выделения специального центрального процессора для оценки действия ресурсного алгоритма;

- проверка ошибок осуществляется в каждом узле, при этом проверяется правильность кода передачи, таблиц передачи и сообщений о следовании.

Организация рестарта в сети также имеет особенности. Различают холодный рестарт, когда игнорируются все таблицы состояний; горячий, когда все таблицы признаются истинными. Рестарт осуществляется загрузкой либо всей сети, либо участка сети.

Как видно из вышеизложенного, специфичные для сети методы противоречивы. Для целей проверки оказывается полезной централизация функций контроля, введение специальных линий и шин связи для информации о состоянии оборудования, хотя при этом снижается надежность сети.

Как представляется, сети должны строиться самопроверяемыми в том смысле, что каждый узел сети должен давать информацию о своем состоянии «исправен или неисправен» по двухшинной связи в виде кодов 10 и 01 (наличие 11 и 00 будет свидетельствовать о неисправности линий связи).

Одним из вариантов может быть шинная или кольцевая структура сети. Общая шина может быть при этом дублированной или троированной.

При организации работы сети важное значение для контроля имеет прогнозирование поведения программных средств, которое дает возможность упростить процесс принятия решения.

Перспективным решением представляется разделение больших сетей на группы узлов, которые осуществляют взаимную диагностику между группами, и разработка различного рода имитаторов для ускоренной отладки и проверки функционирования сетей.

3.4. Классификация и описание методов тестового диагностирования отдельных ЭВМ

Для отдельных ЭВМ в настоящее время существуют те же три основные схемы организации тестового диагностирования, основанные на аппаратной реализации в виде аппаратного ядра, что и для ВС:

- с централизованным встроенным ядром;

- с централизованным внешним ядром;
- с распределенным встроенным ядром.

При использовании централизованного ядра ЭВМ представляется в виде двух частей: S_d, которая соответствует ядру, и S_d, которая соответствует ОД.

Недостатком такой схемы диагностирования является зависимость достоверности принимаемых решений от надежности оборудования ядра. Это обстоятельство приводит к необходимости резервирования ядра, либо организации его по мажоритарному принципу.

Поскольку вычислительные машины как объект диагностирования весьма сложные, приведенные выше схемы тестирования применяют одновременно с разделением машины на блоки. Каждый блок является объектом диагностирования, и в функции ядра входит также обеспечение блокировки воздействий от непроверенных блоков. Стремление к экономии аппаратурных затрат привело к использованию в такой схеме процедур проверки по способу «раскрутки», когда ранее проверенные блоки используются для проверки последующих.

В рамках указанных выше схем тестирования могут применяться различные процедуры диагностирования, среди которых можно выделить:

- процедуру «дерево», при которой поиск начинается с большого числа функциональных блоков и на каждом шаге проверяется «успех» либо «неуспех» прохождения теста, в случае «неуспеха» осуществляется переход к более подробному анализу подозреваемых блоков; эта процедура соответствует реализации условных алгоритмов диагностирования;
- процедуру «начиная с малого», при которой проверяется вначале малая часть, а затем небольшими приращениями остальная часть машины;
- процедуру «пересечения», соответствующую безусловным алгоритмам поиска, при которой после анализа серии результатов тестов выделяются «подозреваемые» на неисправность блоки.

Схема тестового диагностирования предполагает также выбор одного из способов: автоматического, ручного, микропрограммного, подготовки тестовых последовательностей для проверки отдельных устройств ЭВМ и способов построения словарей неисправностей (автоматическим, ручным, смешанным).

В качестве примера на рис.6 показаны схемы тестового диагностирования, применяемые в ЭВМ.

Здесь приняты следующие обозначения: ПМК – память микрокоманд, ВЗУ – внешнее запоминающее устройство, ОЗУ – оперативное запоминающее устройство. Схема на рис.6,*а* соответствует организации тестового диагностирования. Внешний аппаратный тестер, названный консольным файлом, загружает микропрограммную память и начинает микродиагностику. Микродиагностическая программа-резидент контролирует устройство управления и арифметико-логический блок (рис.6,*в*), а остальные микропрограммы проверяют память (рис.6,*б*). Используемая стратегия заключается в тестировании малых схем контроля (СК) (рис.6,*г*), и если перед

обнаружением ошибки персональный компьютер оказался исправным, то он сразу используется для проверки подозреваемого оборудования.

Стратегии использования основного и дополнительного оборудования в процессе тестового диагностирования могут быть разными. Так, в системе применяется следующая стратегия:

- проверяется контрольная аппаратура;
- выбирается один из имеющихся в ЭВМ тестов;
- проверяются схемы сравнения информации;
- проверяется базовая часть ОЗУ;
- производится загрузка в ОЗУ диагностического пакета из ВЗУ.

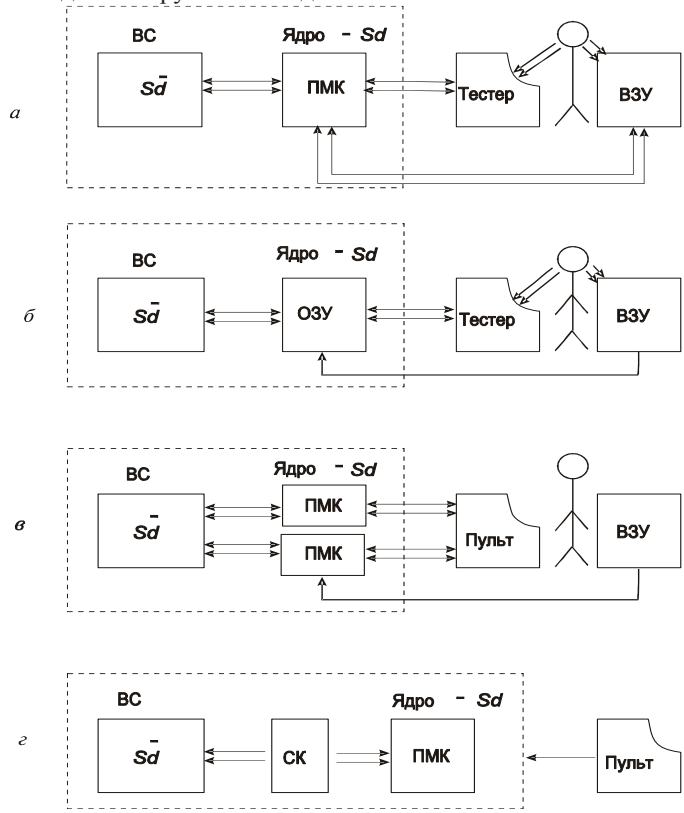

Рисунок 6 – Схемы тестового диагностирования, применяемые в современных ЭВМ

Эти операции проводятся вручную. Затем проводятся обычные операции, основанные на использовании принципа расширяющихся областей. Для этого:

- сравнивается информация, считываемая из устройства управления, с эталонной;

- проверяются служебные блоки ЭВМ с помощью устройства управления;
- проводится программная проверка каналов и терминального комплекта ЭВМ;
- проверяется полный объем ОЗУ тестами из ПЗУ.

Следует заметить, что использование в схемах тестового диагностирования ЭВМ тестов, составленных на уровне микрокоманд, так называемой микродиагностики, дает возможность сократить затраты оборудования на контроль с 50 до 1 % и время диагностирования, а также обеспечить полноту охвата контролем до 95 %. При этом существенно упрощаются требования к системе подготовки тестов. В ряде случаев они оказываются простыми. Наряду с этими преимуществами не требуются большие объемы памяти при организации процесса тестирования. Дальнейшее упрощение проверок достигается применением контролепригодных элементов.

Шинная организация ЭВМ создает предпосылки для несколько своеобразных процедур проведения тестового контроля *по методу расширяющихся областей*. Одна из возможных организаций тестового диагностирования для ЭВМ такого класса, применяемая фирмой Intel, показана на рис.7.

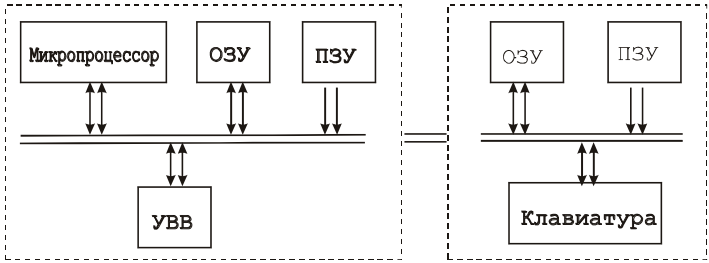

Рисунок 7 – Of-line режим диагностирования

Здесь с помощью клавиатуры устанавливается режим работы, и блок памяти последовательно загружается тестовыми программами. Такой режим диагностирования еще называют of-line. Он обеспечивает обнаружение и поиск относительно простых неисправностей.

Схема организации on-line режима диагностирования изображена на рис.8.

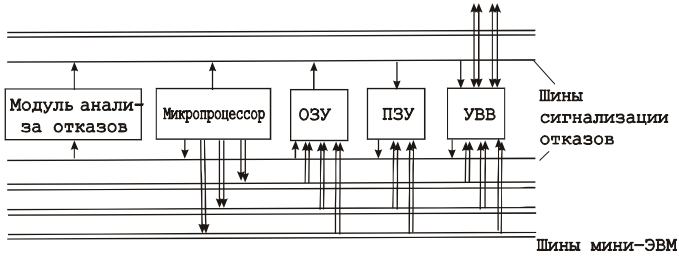

Рисунок 8 – On-line режим диагностирования

Такая организация предполагает дублирование основных блоков и адресных шин, в случае несовпадения выходных сигналов при работе вырабатывается сигнал ошибки, который анализируется специальным устройством.

Из автоматизированных методов проведения проверок ЭВМ следует выделить также *метод анализа логических состояний*. Метод представляет в распоряжение инженеров технические средства - логические анализаторы, которые позволяют проводить измерения и запоминать состояния устройств ЭВМ. Этот метод полезен при нахождении ошибок не только в аппаратном обеспечении, но и программном, поскольку способствует автоматизации процесса отладки программы. Ввиду наличия такого свойства логические анализаторы весьма полезны при отладке систем реального времени, при исследовании последовательных схем. Логические анализаторы дают возможность следить извне, например, за состоянием счетчика микрокоманд, прохождением управляющих команд, за работой системы прерывания и интерфейса, измерять временные параметры. Их использование сокращает время наладки примерно на 20 %.

В настоящее время появились новые контрольно-измерительные приборы, применяемые в качестве сервисного оборудования современных вычислительных машин. Одним из них является анализатор логических состояний. Этот прибор позволяет измерить функциональное состояние испытуемой схемы, индицируя в двоичной форме состояния различных регистров. Другой тип приборов – анализатор логических временных диаграмм. С помощью этих приборов можно воспроизводить на экране обычные временные диаграммы до 12 входных и выходных логических сигналов.

Помимо проверки ВС в целом, существуют схемы организации проверок отдельных блоков. Организация проверок отдельных блоков осуществляется либо на стадии производства, либо на стадии эксплуатации с помощью специального оборудования, не входящего в состав основного оборудования системы.

В настоящее время системы, реализующие эти проверки, должны соответствовать следующим требованиям:

- наличие автоматизированной подготовки исходных данных (тестов, тест-программ и др.);
-возможность использования унифицированного программного обеспечения;
- высокая производительность;
- низкая стоимость;
- простота в управлении и обслуживании;
- возможность применения для новых типов проверяемых устройств.

Рассмотрим основные схемы организации подобных систем. При проверках отдельных схем и элементов обычно выделяют три типа тестирования:

- статическое – частота смены тестовых наборов на входе проверяемого устройства и частота съема реакций значительно ниже, чем при работе устройства в реальных условиях;

- параметрическое - проверяются динамические параметры и предполагается измерение уровней напряжения и тока, задержек и др.;

- динамическое - подаются входные наборы и анализируются выходные реакции проверяемого устройства на частотах, максимальных для данного устройства. Эти типы тестирования могут осуществляться различными средствами. Существующая аппаратура для испытания блоков условно делится на три группы.

К первой группе относятся ручные тестеры, недостатком которых является низкая производительность и невысокая достоверность проверок, обусловленная возможными ошибками человека-оператора (неверная установка теста, пропуск теста, неверный отсчет результата). Эти недостатки особенно проявляются при локализации неисправностей в блоках дискретных устройств. Такие тестеры используются в настоящее время как технологическое и сервисное оборудование.

Ко второй группе относятся автоматические тестеры с элементами программирования. Недостаток устройств этого типа – ориентация на проверку сравнительно несложных блоков.

К третьей группе относятся сложные программные автоматические системы на базе ЭВМ, которые имеют возможность накопления нескольких полных проверочных программ в памяти ЭВМ и проведения серии проверок без вмешательства оператора. Использование ЭВМ дает возможность повысить скорость, точность и достоверность проверок. Эти системы универсальны. Основной недостаток автоматических систем - высокая стоимость, обусловленная использованием дорогостоящего вычислительного комплекса и большими затратами на создание математического обеспечения. Поэтому их более целесообразно использовать в качестве технологического оборудования, поскольку применение для проверки ЭВМ, находящихся в эксплуатации, невыгодно.

При увеличивающейся сложности функциональных блоков ЭВМ программные системы проверки в сочетании с диалоговыми режимами работы пока наиболее эффективны. В связи с этим рассмотрим более подробно их организацию при решении задач статического, параметрического и динамического тестирований.

Типовой вариант проверки статических параметров можно представить структурной схемой (рис.9).

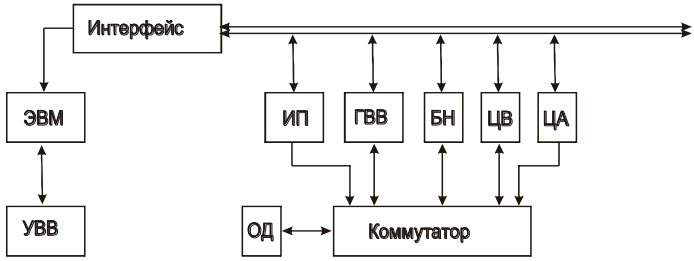

Рисунок 9 - Структурная схема типового варианта проверки статических параметров

В состав системы помимо ЭВМ со стандартными устройствами ввода-вывода (УВВ) входят следующие блоки: источник питания объекта диагностирования (ИП), генератор входных воздействий (ГВВ), блок нагрузки (БН), цифровой вольтметр (ЦВ), цифровой амперметр (ЦА) и матричный коммутатор. Все указанные блоки и приборы связаны магистральной линией через интерфейс с ЭВМ, по которой передается информация из ЭВМ к объекту диагностирования (ОД) и осуществляется ввод информации в ЭВМ. Для функционирования системы достаточно вводить информацию в ЭВМ только с измерительных приборов, однако для обеспечения самоконтроля системы и отдельных ее блоков необходима двусторонняя связь всех блоков с ЭВМ. Разрешение обращения к блокам системы при обмене информацией с ЭВМ поступает по адресной шине. Код адреса может передаваться по магистральной линии связи и тогда дешифрация его осуществляется непосредственно самим блоком либо по радиальной линии связи, и в этом случае функции дешифрации возлагаются на интерфейс.

Типовой вариант схемы автоматической системы тестового контроля (рис.10) отличается от системы параметрического контроля отсутствием в ней измерительных приборов и дополнительным введением блоков, обеспечивающих формирование входных воздействий импульсного и потенциального типов.

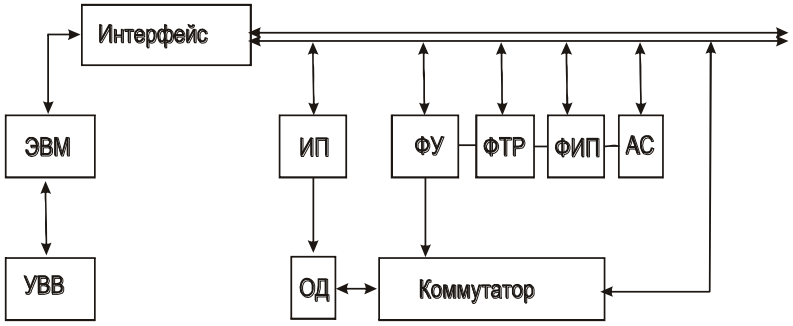

Рисунок 10 – Схема типового варианта автоматической системы тестового контроля

К этим блокам относятся формирователь тестовых наборов и эталонных реакций (ФТР) и формирователь уровней логических сигналов (ФУ). Для фиксации состояний выходов ОД в заданные моменты времени используется анализатор уровней логических сигналов (АС), который стробируется импульсами, вырабатываемыми формирователем импульсов прибора (ФИП). Наличие ФИП требуется также для обеспечения контроля функционирования ОД, так как при этом на входах и выходах ОД необходимо согласовывать определенную временную диаграмму моментов подачи входных воздействий и фиксаций в АС выходных логических сигналов.

Важной характеристикой систем тестового контроля является число информационных выводов в системе, т. е. число каналов для подключения к системе ОД. При этом имеется в виду, что каждый информационный вывод ОД может быть как его входом, так и выходом, к которому должен подключаться АС. Следовательно, все информационные каналы системы должны быть идентичны по выполняемым функциям и число их в системе должно однозначно соответствовать числу информационных выводов ОД.

Такой подход в организации систем тестового диагностирования дает возможность оперативно расширять число информационных каналов системы. Функции переключателя вход–выход выполняет коммутатор. Для проведения операций контроля необходимо на информационные входы ОД передавать тестовые наборы с заданной частотой в определенной последовательности. В настоящее время для этого используются два "варианта аппаратурной реализации блоков формирования и генерирования входных тестовых наборов (выбор того или иного варианта реализации блоков зависит в основном от назначения систем).

Первый вариант предполагает использование буферного запоминающего устройства (БЗУ). Второй вариант аппаратурной реализации устройства формирования тестовых наборов менее универсален, но позволяет значительно упростить аппаратурную часть систем за счет исключения блоков памяти БЗУ. Этот вариант широко применяется при реализации автоматических систем, ориентированных на контроль ограниченного класса ОД, идентичных по функционированию. Примерами таких ОД можно назвать ОЗУ, регистры, счетчики и т. д. В таких системах блоки формирования одновременно выполняют функции генерирования контрольных или диагностических тестов по заданным алгоритмам, а в задачи ЭВМ входит только управление этими блоками.

Автоматическое тестовое оборудование предназначено для выполнения двух основных функций: контроля «исправное» и по принципу «годен – негоден» локализации места неисправностей. В ряде случаев такие установки обладают функциями восстановления. В соответствии с возможностью решения задач обнаружения и локализации можно выделить словарные и зондовые организации поиска неисправностей. *Словарная организация*, как правило, предусматривает автоматический режим диагностики. В простейшем

случае смысл словарного поиска состоит в следующем. Определяется реакция устройства на выбранный набор тестовых сигналов. При проведении испытаний теми же тестами реального устройства реакция сравнивается с полученной информацией и по совпадению выполняется идентификация неисправностей. В непосредственном виде такой подход применяется редко, и в основном в ЭВМ, поскольку хранение полных выходных векторов требует большой памяти (число неисправностей и тестов велико). Для сокращения объема словарей используются методы сжатия двоичной информации.

Зондовая организация поиска (одно- и многоконтактная) предусматривает полуавтоматический диалоговый режим проверки с участием инженера-оператора. Поиск неисправности осуществляется в статическом режиме, анализируется соответствие выходных реакций эталонным: если микросхема исправна, определяются питающие ее микросхемы, с которых поступает сигнал, отличный от требуемого. Для реализации этого режима в конкретной системе необходимо иметь структурное описание схемы, библиотеку логических функций различных типов микросхем, входные и выходные эталонные тактовые сигналы и, кроме того, значения сигналов на всех полюсах исправной схемы при подаче данного входного набора. Этот способ проверки используется как для всего блока, так и для отдельных его элементов. Поэтому различают поэлементный и поблочный контроль. Недостаток зондовой организации проверок заключается в необходимости обеспечения хорошего контакта в месте соединения зонда и исследуемого элемента.

3.5. Задачи отладки АСУ ТП

Организация автоматизированного процесса отладки АСУ ТП является чрезвычайно важной и имеет много общего с организацией процессов диагностирования технических систем. Это решение тех же вопросов: подготовка входных воздействий, выбор точек подачи входных воздействий, выбор способа подачи входных воздействий, выбор способов съема выходных реакций, выбор способов имитации входных воздействий. Причем решение этих вопросов желательно проводить на ранних стадиях создания систем.

Общая блок-схема отладки АСУ ТП показана на рис.11, где ИК и УК – измерительный и управляющий комплексы, ПО – программное обеспечение, ОП – обслуживающий персонал.

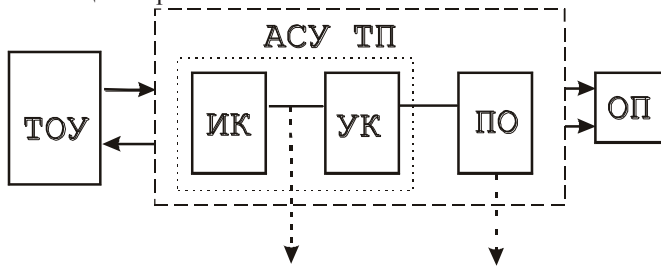

Рисунок 11 – Общая блок-схема отладки АСУ ТП

Обычно используют отладку по функциям, для выполнения которых предназначены АСУ ТП, например, отладку измерительного комплекса, отладку управляющего комплекса, обучение обслуживающего персонала. Опыт отладки существующих АСУ ТП показывает, что наибольшая часть ошибок проявляется на заключительной стадии, а именно при внедрении.

С учетом сложности процесса отладки его, как и обычный процесс проверки, организуют по принципу от «простого к сложному», т. е. путем постепенной отладки и наращивания состава взаимодействующих средств и решаемых задач.

Выделяют два основных этапа отладки: автономный и комплексный. При автономной отладке устраняются ошибки, которые не позволяют отдельным элементам системы функционировать самостоятельно. Иными словами, при автономной отладке устраняются внутренние ошибки, ошибки отлаживаемого элемента.

Уровень отлаженности, определяемый как отношение числа ошибок, выявленных при автономной и комплексной отладках, составляет 0,3–0,6.

При комплексной отладке проводится проверка взаимодействия технических средств и программного обеспечения при выполнении основных функций контроля и управления. Уровень отлаженности составляет 0,8–0,9.

С точки зрения автоматизации процессов отладки наиболее просто осуществляется автономная отладка: можно легко автоматизировать подачу и съем воздействий, построить имитаторы.

Наиболее типичные ошибки, которые встречаются во время отладки, делятся на три группы:

- алгоритмические, связанные с некорректной постановкой задач контроля и управления, неучетом «второстепенных» факторов, превышением выделенных ресурсов времени, памяти и др.;
- системные, определяемые уровнем соответствия требуемого и фактического порядка количества информации между отдельными устройствами (подсистемами) в составе системы;
- схемные и программные, зависящие от уровня организации работ, квалификации исполнителей, уровня технологии и надежности технических средств. Наибольшую опасность представляют ошибки первой группы, что требует разработки средств контроля этапа алгоритмического проектирования.

Важное значение имеет разработка такой структурной и функциональной организации АСУ ТП, которая обеспечивала бы удобства для автоматизированной отладки. Иными словами, создаваемый комплекс должен быть в максимальной степени пригоден для отладки на всех стадиях своего существования, с тем, чтобы на реальном объекте не затрачивать значительные ресурсы и время.

Важнейшей задачей ранних этапов является анализ технического задания на полноту, выполняемость, непротиворечивость. Автоматизация такого

анализа дает возможность снизить долю ошибок, обнаруживаемых на более поздних этапах создания систем. Решение этой задачи связано с математическим моделированием всевозможных условий функционирования в АСУ ТП, в поиске состояний АСУ ТП, нормальное функционирование которых связано с реализацией функций, не вошедших в техническое задание (ТЗ). Анализ на противоречивость сводится к поиску ситуаций, приводящих к противоречиям в ТЗ. Анализ на выполняемость предполагает оценку технических возможностей реализации существующими средствами. Эти виды анализа удобно реализуются с помощью имитационных моделей.

При непосредственной отладке АСУ ТП могут использоваться физические и имитационные модели либо их различные сочетания, реализуемые с помощью универсальных ЭВМ или специализированных средств.

Сочетания моделей используют при комплексной отладке ВС. Для этой цели используются специальные аппаратные имитаторы внешних условий, заменяющие реальный объект, а также применяются соответствующие средства анализа. Структура системы отладки в значительной мере зависит от типа используемой ЭВМ и имеющихся в ее составе средств отладки программ.

Обычная автоматизированная система отладки включает две связанные машины: одну, на которой выполняется комплекс отлаживаемых программ, т.е. обработка результатов отладки в реальном времени, и вторую, управляющую, на которой производится имитация реальных условий.

Такая система полезна не только для комплексной отладки программ, но и для отладки отдельных частных программ, так как позволяет использовать мощные средства инструментальной машины.

Физические модели широко используются для отладки программ систем, разрабатываемых на основе микропроцессоров. В таких системах имеется физическая модель системы, представляющая собой мини-ЭВМ, затем в нее вставляется отлаживаемый блок и проводятся отладочные операции. В качестве средства анализа широко используются логические анализаторы. Последние позволяют проследить путь программы, остановить ее выполнение в нужном месте, потактово выполнять отдельные команды.

Таким образом, процесс отладки сложных систем также нуждается в специальных методах его эффективной организации, количественных оценках для выбора рациональных вариантов, разработке новых способов и методов организации проверки. Как представляется, здесь с успехом могут быть применены принцип декомпозиции и агрегированные описания сложных систем.

3.6. Автоматизация процесса диагностирования сложных технических объектов

Сложность современных объектов диагностирования приводит к использованию иерархических структур системы диагностирования (рис.12).

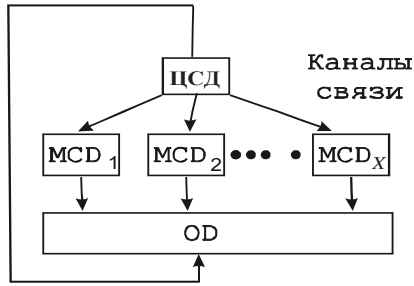

Рисунок 12 – Схема 1 иерархических структур системы Диагностирования

На первом уровне такой системы имеются местные системы диагностирования блоков, на втором – центральные системы диагностирования (ЦСД). Причем не обязательно создание специальных систем диагностирования, их функции можно выполнять комплексом технических средств, находящихся в составе существующих и разрабатываемых АСУ ТП.

К объектам такого класса можно отнести автоматизированные системы отправки грузов, где важно своевременное установление факта неисправности.

Иерархические системы диагностирования могут иметь различные функции на каждом уровне иерархии. Например, на первом уровне может осуществляться только обнаружение неисправностей, а на втором – их локализация. Распределение функций по уровням иерархии может быть смешанным. Во всех случаях необходимо располагать определенными правилами получения и обработки диагностической информации.

К особенностям диагностирования сложных систем относится необходимость передачи диагностической информации на значительные расстояния, что требует применения в их составе специальных телекоммуникационных средств.

Общим для всех систем иерархического типа является наличие «инженерного пульта» системы, на котором отображается сводная информация о состоянии значительного числа технических устройств с целью своевременного принятия решения по их эксплуатации.

На рис.13 показана иерархическая система диагностирования технического объекта, построенная с использованием микропроцессоров. В этой системе отдельные участки объекта диагностирования проверяются собственными микропроцессорами, а оценка результатов проводится централизованно. Для организации связи между микропроцессами могут использоваться независимые шины и общая шина.

Возможны другие типы организации иерархических систем диагностирования на микропроцессорах. Например, если не требуется высокое

быстродействие системы и имеется почти автономный режим работы, то вместо верхнего микропроцессора может использоваться общая оперативная память, через которую осуществляется обмен информацией

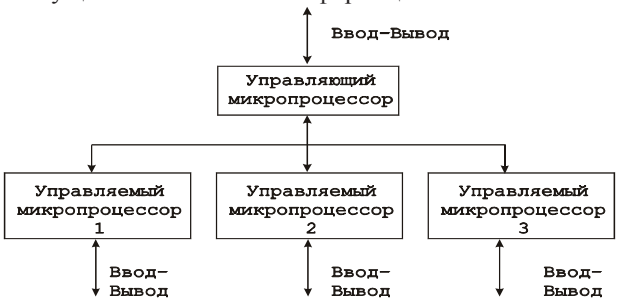

Рисунок 13 – Схема 2 иерархических структур системы диагностирования

3.7. Диагностические модели

Моделью называется объект любой природы, который способен замещать исследуемый объект так, что его изучение дает новую информацию об этом объекте. Математические модели описывают математической символикой разнообразные связи между входными, выходными и внутренними параметрами в различных сочетаниях. В качестве диагностических моделей могут рассматриваться дифференциальные уравнения, логические соотношения, диаграммы прохождения сигналов и др.

Первостепенное значение имеют структурные свойства объекта диагностирования. Эти свойства определяются совокупностью элементов и связей между ними, формируются при создании системы и проявляются в процессе его работы. Главная роль в формировании любой системы, а следовательно, и в характеристике ее структурных свойств принадлежит межэлементным связям, поскольку именно они придают системе целостный характер и обеспечивают взаимодействие элементов. Выявление вида связи между элементами позволяет дифференцировать систему на ряд уровней с одинаковой природой связи, найти форму ее использования через измеряемые параметры, определить характер изменения последних и сформулировать задачу диагностирования, свойственную подсистеме одного уровня.

При проектировании ВС разрабатывают диагностические модели объектов. Формальное описание объектов, учитывающее возможность изменения его состояния со временем, т.е. исправного и неисправного состояний, называют *диагностической моделью.*

Различают модели:

- формализованные модели - описание объекта в аналитической, графической, табличной или другой форме;

- явные модели - содержат наряду с описанием исправного объекта описание каждой из неисправных модификаций;

- неявные модели - содержат только описание, например, исправного обьекта;

- функциональные модели - содержат описание выполняемых исправным и неисправным объектом функций. Данные модели позволяют решать задачи проверки работоспособности и правильного функционирования объекта;

- структурные модели (модели технического состояния) - содержат информацию о внутренней организации объекта, о его структуре;

- детерминированные модели - диагностический сигнал описывается детерминированными функциями;

- вероятностные модели - для описания поведения объекта используют статистические параметры;

- логические модели - при большом числе структурных и диагностических параметров для их анализа используется аппарат математической логики.

По методам представления взаимосвязей между состоянием объекта, его элементами и параметрами, диагностические модели подразделяют на следующие виды: непрерывные, дискретные, специальные.

Диагностические модели нужны для построения алгоритмов диагностирования формализованными методами. Другим важным назначением моделей объектов является их применение для формализованного анализа заданных (в том числе построенных интуитивно, вручную) алгоритмов диагностирования на полноту обнаружения, на глубину поиска дефектов или на предмет построения *диагностических словарей.*

Диагностические словари - это один из способов сжатия информации с тем, чтобы получить документацию, удобную для пользования. Как правило, диагностические словари состоят из двух колонок, в одной из них указываются симптомы проявления неисправностей, а в другой - ассоциированные с ними неисправности.

Диагностическим (контролируемым) параметром называется параметр объекта, используемый при его диагностировании (контроле) [2].

Выделяют *рабочее техническое диагностирование* - диагностирование, при котором на объект подаются рабочие воздействия и *экспресс-диагностирование* - диагностирование по ограниченному числу параметров за заранее установленное время.

Для описания процессов диагностирования трудно учесть все факторы, характеризующие эти процессы, поэтому применяются различные математические модели, описывающие отдельные свойства ВС. В настоящее время для расчета вероятностных характеристик используются методы случайных процессов, сетевые методы и др. Для описания структурных свойств применяются теоретико-графовые методы (например, методы, основанные на представлении структуры системы в виде мультиграфов и взвешенных графов). Для учета функционального поведения используются

логические схемы алгоритмов, микропрограммные алгебры, операторные схемы.

Разнообразие конфигурации управляющих систем затрудняет создание комплекса достаточно универсальных математических моделей, пригодных для широкого распространения. В связи с этим удобно применять декомпозиционный прием, при котором выделяются типовые схемы организации проверок отдельных блоков, например вычислительных систем.

Применение этих моделей дает возможность описывать процессы оперативного обнаружения, поиска и устранения неисправностей, реализуемые аппаратными и программными средствами в вычислительных машинах и системах. Для расчета сложных конфигураций используются различные сочетания базовых моделей, что существенно упрощает процедуру оценки вариантов.

Аппаратно-программное обеспечение для решения задач автоматизации СД может быть универсальным и специализированным. Универсальное обеспечение реализуется в виде пакетов прикладных программ для решения указанных задач, написанных на языках высокого уровня. Специализированное аппаратно-программное обеспечение представляет собой реализацию на специализированных языках автоматического проектирования.

Возможны также другие принципы организации аппаратно-программного обеспечения. Наличие теории организации процессов диагностирования позволяет проводить обоснованный выбор вариантов систем диагностирования, сократить сроки и улучшить качество проектирования путем использования типовых специальных правил и алгоритмов и применения их в автоматизированных системах проектирования.

Системам диагностирования сложных объектов присущи все черты сложных систем: многообразие структуры, многосвязность; возможность изменения состава и состояния, многообразие природы элементов, многокритериальность и многомерность.

Задача синтеза структуры системы диагностирования (СД) включает в себя: выбор состава контрольных точек для подключения системы диагностирования; выбор принципа построения системы диагностирования; распределение проверочных функций по узлам и блокам; выбор принципов принятия решения; выбор методов реализации проверенных функций и согласования их с целями, общими для системы; назначение состава технических средств и т. д.

Будем полагать, что существует целевая функция Φ, характеризующая качество выполнения основной цели диагностирования. Обозначим через y вариант построения системы диагностирования, через Y – множество таких вариантов.

Тогда в общем случае задача выбора рационального варианта СД состоит в нахождении такого $y_0 \in Y$, чтобы $\Phi(y_0) \to \text{extr}$.

Если рассматривать процесс проектирования более подробно, то y_0 определяется принципом организации π и набором технических средств $X = \{x_1, x_2, x_3, \ldots\ldots, x_n\}$.

Целевая функция процесса диагностирования может выражаться по-разному, в виде функции потерь функционирования объекта, обусловленных его неидеальностью, либо в виде частных показателей, характеризующих отдельные операции общего процесса диагностирования – достоверностью, временем поиска и т. д.

Достижение определенных показателей целевой функции сопряжено с дополнительными затратами материальных ресурсов: оборудования, памяти, времени и т. д.

Исходя из этих соображений, может быть достаточно большое число формальных постановок задач, при которых выбирается оптимальный состав оборудования. Для их решения наибольший интерес представляет разработка процедур, ориентированных на ранние стадии проектирования.

Рассмотрим две методики верхнего уровня, ориентированные на выбор принципов организации систем диагностирования. Эти методики должны обеспечивать:

- генерацию множества допустимых альтернативных вариантов систем диагностирования;
- оценку альтернативных вариантов и выбор наиболее перспективных.

Способы генерации вариантов и методы их отсева определяют эффективность процесса формализации, поэтому поиск удачных решений - весьма актуальная задача.

Рассмотрим вариант процедуры верхнего уровня, когда связь с параметрами СД и характеристиками ее качества устанавливается в явном виде и может быть описана, например, расчетными соотношениями теории надежности. Методика приведена на рис.14 и включает следующие этапы:

1) Оценивается степень соответствия достигаемых показателей надежности и эффективности заданным в технических требованиях. По результатам оценки принимается решение об использовании дополнительных средств в системе диагностирования;

2) Выделяются подсистемы комплексной системы диагностирования: обнаружения, поиска и устранения неисправностей, а также накопления статистических данных об отказах;

3) Составляются уравнения связи между основными параметрами подсистем диагностирования и характеристиками надежности и эффективности систем;

4) Определяются формализованные технические требования к характеристикам подсистем СД на основании решения уравнений связи и заданных технических требований к СД;

5) Для каждой подсистемы СД формулируются оптимальные задачи по выбору состава контрольного оборудования и точек его размещения.

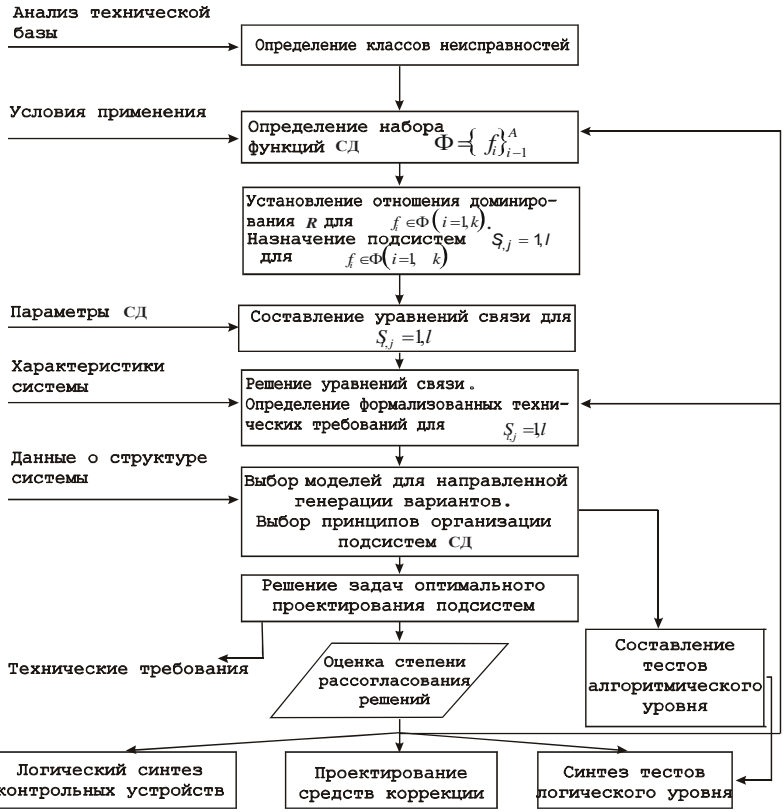

Рисунок 14 – Структуры систем диагностирования

Определяются критерии выбора и ограничения, налагаемые на отдельные параметры подсистем;

6) Решаются оптимальные задачи в соответствии с п.5. При этом генерируются различные альтернативные варианты структур, оценивается их качество и производится отсев нерациональных вариантов;

7) Проводится согласование полученных частных решений для подсистем с целью объединения их в СД. Проводится корректировка с целью исключения избыточного оборудования;

8) Оценивается влияние характеристик спроектированной СД на показатели надежности и эффективности с учетом выделенных классов неисправностей. По результатам оценки проводится корректировка состава технических средств в СД.

На рис.15 изображено прохождение информации согласно этой методике.

Возможен другой вариант методики верхнего уровня, который применяется в случае отсутствия в явном виде зависимости между показателем эффективности СД и характеристиками средств контроля. Он основан на использовании процедуры двухэтапной оптимизации. Второй вариант включает следующие этапы:

- выделяются подсистемы $S_i \quad i = \overline{1,k}$;
- вводится критерий качества для S_i, $F_i(x_j)$, x_j – вектор состава;
- решается задача нахождения x_{i0} такого, что $F_{i0}(x_i) \to \max$;
- составляется формула

$$W(x,\lambda) = \max \lambda^i \left(\frac{F_{i0} - F_i(x)}{F_{i0}} \right),$$

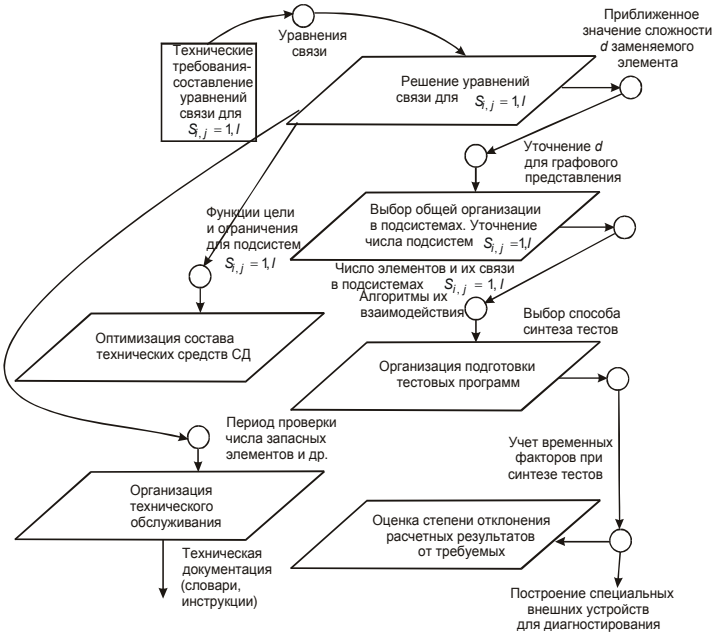

Рисунок 15 – Схема прохождения информации в системе диагностирования

где λ^i — вектор концепции, $\lambda^i \geq 0$; $\sum_{i=1}^{n} \lambda^i = 1$, F_{i0} – идеальное значение критерия качества;

- фиксируется λ^i;
- решается задача $W(x,\lambda) \to \max$;
- строится $W(\lambda) = W(x,(\lambda),\lambda)$ для любых λ;
- находится $\Phi(\lambda) = \Phi(x(\lambda)) \to \max$;

- находится λ, а затем X.

Генерация вариантов СД является основным этапом в процессе синтеза ее структуры и формализация этого процесса сложная. Поэтому далее его будем рассматривать применительно к этапу, когда система диагностирования разделена на подсистемы $S_i \quad i=\overline{1,n};$, а формализованное задание вариантов осуществляется в рамках каждой подсистемы.

Каждая подсистема S_j предназначена для выполнения определенной задачи $З_j$. Выполнение задачи $З_j$ можно представить определенной совокупностью операций O_j, $j=\overline{1,k}$.

Совокупность операций O_j, $j=\overline{1,k}$ применяется для всего объекта диагностирования либо ввиду его сложности для совокупности его блоков S_j. Таким образом, осуществляется отображение функций O_j, $j=\overline{1,k}$ на блоки объекта диагностирования.

В свою очередь, для реализации каждой из операций O_j можно использовать различные технические средства x_{ij}. Например, в простейшем случае операцией является оперативное обнаружение ошибок, а реализуется она совокупностью различных средств и методов проверки.

Для формального описания операций можно применять известные математические описания: граф-схемы, микропрограммные алгебры и др.

Возможные варианты построения СД формируют путем использования морфологических блоков. Морфологическим блоком средств проверки называется матрица, в которой столбец соответствует различным типам элементарных средств проверки, а строка - различным вариантам средств проверки одного типа. Элементарным средством проверки называется техническое устройство, выполняющее элементарную функцию в процессе проверки. Морфологические блоки строятся для каждой подсистемы S_j. В рамках выделенной подсистемы S_j морфологические блоки строятся для отдельных блоков объекта диагностирования.

Вариант технической реализации формируется следующим образом. Из каждого морфологического блока выбирается по одному элементу и образуется вектор состава $y=\{x_1,\ldots,x_n\}$, который будет фигурировать в дальнейшем изложении.

Необходимо отметить, что даже в такой постановке число вариантов весьма велико, что требует использования формальных методов оценки и отсева нерациональных вариантов.

Следует отметить имеющуюся возможность усечения вариантов за счет доминирования одних подсистем над другими. Так, в различных подсистемах S_j общей системы диагностирования требуется реализация некоторого числа сходных функций. Поэтому после выбора варианта, например, для S_j, можно не рассматривать для S_{i+1} варианты реализации функций, которые предусмотрены вариантом для S_j.

Такое свойство характерно для подсистем обнаружения и поиска неисправностей, когда средства обнаружения можно использовать в виде дополнительных контрольных точек при поиске неисправностей.

Составление уравнений связи между подсистемами основывается на следующем. Каждой подсистеме S_j, $i=\overline{1,n}$ можно поставить в соответствие один или несколько критериев качества $\Phi_j(S_i, y_i)$, $j=k$, где y_i- вектор состава средств проверки, для получения которых требуются соответственно материальные ресурсы $C_j(\Phi_j, y_i)$.

Объект и средства диагностирования в процессе функционирования могут находиться в различных отказовых ситуациях p_z, $z=\overline{1,l}$, характеризуемых определенными весами d_z. Вероятности нахождения в z-м состоянии являются некоторыми функциями от $\Phi_j(S_i, y_i)$.

Тогда связь между параметрами системы диагностирования можно установить следующим путем: $\max \sum_{z=1}^{l} p_z d_z$ при $\sum_{i=1}^{n} C_i(\Phi_i, y_i) \le C$.

Здесь C - допустимые затраты материальных ресурсов на создание системы диагностирования. Из решения этой задачи находятся значения Φj, которые удовлетворяют ограничениям и служат техническим заданием на проектирование системы диагностирования.

Для каждого из случаев применения информационно-вычислительных систем могут быть свои критерии эффективности и свои специфичные уравнения связи. Например, пусть требуется выбрать оптимальный вариант СД по критерию средних потерь, представленному в виде

$$\Phi_2 = \lambda P\left[C_1 + C_2\left(T_1 + T_2 R\right)\right]t + \lambda\left(1-P\right)t\left[C_1'' + C_2'' T_a\right],$$

при ограничении

$$D_1(P) + D_2(R) = \tilde{D},$$

где λ – интенсивность потока ошибок всей УВС, P - вероятность обнаружения ошибок подсистемой оперативного обнаружения ошибок (ПОО),

R- разрешающая способность подсистемы поиска неисправностей (ППН),

C_1, C_2 - потери в случае обнаруженного отказа,

C_1', C_2' - потери в случае пропущенного отказа,

T_1, T_2, T_α - времена устранения отказов, t - время работы системы. Представим Φ_2 в виде

$$\Phi_2 = \lambda\left(C_1' + C_2' T_a\right)t + P\,\lambda\, t\left\{\left[C_1 + C_2\left(T_1 + T_2 R\right)\right] - \left(C_1' - C_2' T_a\right)\right\}.$$

Ограничения с учетом сделанных ранее предпосылок представим таким образом

$$\alpha_1 + \alpha_2\beta + \beta_1 + \frac{\beta_2}{R} = \tilde{D}.$$

Для нахождения минимума функционала Φ_2 воспользуемся методом неопределенных множителей Лагранжа. Тогда получим систему уравнений

$$\frac{d\bar{\Phi}_2}{dR} = \lambda\, t\left\{\left[C_1 + C_2\left(T_1 + T_2 R\right)\right] - \left(C_1' - C_2' T_a\right)\right\} + \lambda\,\alpha_2 = 0,$$

$$\frac{d\bar{\Phi}_2}{dR} = P\,\lambda\, t\, C_2 T_2 - \frac{\beta^2}{R^2} = 0,$$

$$\alpha_1 + \alpha_2 P + \beta_1 + \frac{\beta_2}{R} - \tilde{D} = 0,$$

где $\bar{\Phi}_2 = \Phi_2 - \lambda\left[D_1(P) + D_2(R) - \tilde{D}\right]$.

Из решения этой системы находится неопределенный множитель λ и оптимальные значения $\breve{P}$ и $\breve{R}$. Если оказывается, что $\breve{P}$ и $\breve{R}$ лежат вне области их определения, то ищутся ближайшие к ним на границах интервала определения. Оптимальные значения $\breve{P}$ и $\breve{R}$, полученные в обоих случаях, служат заданием на проектирование ПОО и ППН.

Полученные при решении уравнений такого типа технические требования могут быть затем пересчитаны для отдельных подсистем.

Рассмотрим теперь определение такого разбиения структуры ЭВМ, при котором достигается экономия затрат при сохранении других показателей в заданных пределах.

Как отмечено выше, качество поиска характеризуется разрешающей способностью *R*, определяемой как число подозреваемых на неисправность сменных элементов машины (в первом приближении можно пользоваться средним числом *R*).

Из определения *R* вытекает связь между ППН и подсистемой устранения неисправностей (ПУН): чем меньше значение разрешающей способности *R* ($1 < R < n$, n - число элементов замены), тем меньше требуется элементов замены.

В то же время высокие показатели разрешающей способности достигаются, с одной стороны, использованием тестов, имеющих улучшенные характеристики по сравнению с имеющимися, либо использованием специальных схемных решений, что связано с дополнительными затратами оборудования.

Опишем формальную постановку задачи оптимизации для частных подсистем и их сетевая интерпретация. Рассмотренные выше методики предполагают многократное решение задачи максимизации функционала $\Phi(x)$ при определенных ограничениях. Желательно, чтобы решение было наглядным и проводилось достаточно быстро. Многие задачи оптимизации состава оборудования подсистем общей СД сводятся к минимизации подобного функционала.

Рассмотрим некоторые наиболее часто встречающиеся задачи оптимизации для информационных систем. Обычно для информационных

систем общая система диагностирования содержит подсистему обеспечения достоверности, подсистему поиска неисправностей, подсистему управления.

Достоверность функционирования сложной системы, состоящей из n блоков, определяется по формуле

$$R(t,Y)=\prod_{i=1}^{n}R_i(t,y_i),$$

где $R_i(t,y_i)$ - достоверность функционирования i-го блока

$Y=\{y_i\}$, $t=\overline{1,n}$ - вектор состава средств обеспечения достоверности, n-число блоков. Под блоком можно условно понимать этап обработки, устройство, программный модуль.

Тогда задача выбора оптимального состава средств обеспечения достоверности (иначе - подсистемы оперативного обнаружения ошибок) сводится к нахождению $Y_0 \in Y$ такого, что

$$R(t,Y_0)\to \max$$

при $C(Y_0)$, $x_j \le C_0$, $\sum_{j=1}^{k} x_j = 1$,

либо к нахождению вектора Y_0 такого, что $R(i,Y_0)=\sup R(t,Y)$ (Y – множество допустимых решений), т. е.

$$C_k(Y)\le C_k^0$$

Здесь $C(Y_0)$ - затраты на выбор средств обеспечения достоверности; C_0 – допустимые затраты; x_j - двоичная переменная, принимающая значение 1, если j-й блок проверяется j-м способом, и 0 – в противном случае; k – число способов проверки.

Для случая нескольких ограничений задача формулируется таким образом: обеспечить максимум достоверности при ограничениях на дополнительные затраты оборудования, времени и т. д. Иными словами, это означает найти вектор Y_0 такой, что

$$R(t,Y_0)=\sup_{Y_0\in Y}(t,Y)$$

где Y - множество всех допустимых решений.

Кроме приведенных ограничений (будем называть их методическими), существует ряд ограничений, которые необходимо отметить. Системы диагностирования могут строиться с использованием рассредоточенного или централизованного принципа, причем возможен также вариант сквозного, нигде не прерываемого контроля, при котором контролируются либо все блоки ЭВМ, либо их большая часть и возможно наличие уже готовых решений, приоритетных и т. д.

Аналогично задача обеспечения тестового диагностирования (выбора подсистемы поиска неисправностей (ППН)) сводится к распределению дополнительных контрольных блоков по структуре системы. Тогда можно требовать минимазации $min\ П(Y)$ при $C(Y_0)\le C_0$, где $П(Y)$ - показатель

качества ППН; Y – вектор состава, $C(Y_0)$ -дополнительные затраты; C_0 – допустимые затраты.

Если средством устранения является резервирование и ЗИП, то задача выбора их состава тоже сводится к подобной постановке.

При решении указанных задач удобна сетевая интерпретация возникающих задач синтеза. Суть сетевой интерпретации задачи выбора состава контрольного оборудования для отдельных подсистем состоит в следующем. Соответственно функциональным блокам проверяемого объекта и определенным микрооперациям строится вариантный граф $G(E,U)$, каждой вершине $e_i \in E$ которого ставится в соответствие способ проверки i-го блока, а дугам $u_i \in U$ соответствуют возможные сочетания между ними. Дуги графа $G(E,U)$, «взвешиваются» весами: достоверностью функционирования, интенсивностями отказов, затратами оборудования и времени на реализацию контроля.

Введем две фиктивные вершины e_0 и e_k (соответственно начальная и конечная), которые соединяются со всеми вершинами подмножеств $E_0 \in E$ и $E_k \in E$, отвечающих вариантам проверки начального и конечного блоков проверяемого объекта.

Оптимальной организации соответствует кратчайший допустимый путь в графе $G(E,U)$. Алгоритмы решения этой задачи при нескольких ограничениях легко реализуются на ЭВМ.

Рассмотрим общую процедуру рациональной организации системы тестового диагностирования (ТД). Эта методика состоит в следующем.

На первом шаге определяется предварительная организация подсистемы ТД на основе решения уравнений связи, связывающих параметры ТД с характеристиками эффективности системы, определяется централизованный либо распределенный принцип размещения диагностического оборудования (выделяется одно «ядро» или несколько «ядер» системы).

На втором шаге используется представление системы S в виде взвешенного направленного графа $G(E,U)$, каждой вершине которого соответствует функциональный узел, а дугам - связи между ними.

На третьем шаге определяется связность графа. Кроме того, на графе $G(E,U)$ отмечаются точки съема и подачи контрольной информации, имеющиеся в S (в силу естественной структурной и функциональной избыточности) . Полученные точки разделяют граф $G(E,U)$ на некоторые части. $G_i^n(E_i^n, U_i^n)$.

Полученные части взвешиваются: вершинам ставится в соответствие количество оборудования узлов, а дугам – число связей между узлами.

На пятом шаге с помощью специального алгоритма (Форда – Фалкерсона) части графа $G_i(V_i, U_i)$ разрезаются на более простые части $G_i(E_i, U_i)$.

На шестом шаге по полученным разрезам E_k проводится размещение дополнительного контрольного оборудования, облегчающего проведение процедур проверки: блокирующих вентилей, генераторов тестов, схем сравнения и др.

Для окончательного выбора структуры ТД на седьмом шаге составляется вариантный граф *G(V,U),* по следующим правилам: каждому разрезу E_k ставится в соответствие множество V_k способов организации проверки узлов, входящих в состав подграфа, определенного разрезом; дуги графа *G(V,U)* соответствуют сочетаниям различных способов проверки узлов, входящих в подграфы $G_i^n(E,U)$. Дугам присваиваются «веса»: дополнительные затраты оборудования и времени, разрешающая способность и достоверность поиска.

На восьмом шаге оптимальному варианту ТД соответствует вектор состава $Z=\{z_1,z_2,\ldots,z_i,\ldots,z_k\}$, при котором достигается максимум разрешающей способности и достоверности поиска при ограничении на дополнительные затраты оборудования и времени.

На заключительном шаге проектируется программное обеспечение для организации взаимодействия выбранных контрольных блоков.

Реализация этих процедур применительно к сложным системам осложняется высокой размерностью возникающих графовых задач, что требует разработки специальных алгоритмов упрощения и сокращения размерности.

Для указанных постановок задач существенным является выбор ограничений - дополнительных затрат.

3.8. Специализированные алгоритмы и программные средства диагностики, используемые в гражданской авиации

Разработка перспективных бортовых цифровых вычислительных систем (БЦВС), средств ВТ в классе структур интегрированной модульной авионики (ИМА), предназначенных для эксплуатации в ГА, сопряжено с необходимостью разрабатывать специализированные алгоритмы и программные средства контроля технического состояния аппаратуры [6]. Алгоритмы контроля гарантируют заданную полноту и достоверность проверки при проведении этапов тестирования мультипроцессоров и их компонентов на заводе-изготовителе и в эксплуатации.

Бортовая цифровая вычислительная система класса ИМА представляет собой интегрированную вычислительную платформу, в состав которой входят:

- аппаратные средства: быстросменные конструктивно-функциональные модули (КФМ): модули вычислительные МВ, модули ввода-вывода МВВ, модули массовой памяти ММП, модули напряжений МН, модули графические МГ - устанавливаемые в типовую несущую конструкцию (крейт);
- программные средства: функциональное программное обеспечение (ФПО), операционная система, поддерживающие программные средства - управляющие ресурсами БЦВС в целях создания реконфигурируемой вычислительной среды.

В [9] рассмотрены обобщенные алгоритмы тестирования БЦВС класса ИМА Для проверки функционирования БЦВС проводится тестовый контроль аппаратуры. При эксплуатации контроль осуществляется средствами встроенного контроля. Каждый модуль БЦВС имеет встроенные средства контроля. Встроенный контроль входит в состав базового программного обеспечения (ПО) каждого модуля. Для полной диагностики БЦВС используется специализированное ПО, которое обрабатывает сигналы состояния исправности отдельных модулей и формирует интегральный сигнал исправности БЦВС. Определение технического состояния БЦВС при эксплуатации осуществляется средствами встроенного контроля (аппаратными и программно-логическими). Программно-логическими средствами контроля проводятся тесты: начального включения каждого входящего в БЦВС модуля; фонового контроля модуля во время загрузки ПО; встроенного контроля каждого входящего в БЦВС модуля.

Проверка БЦВС при эксплуатации осуществляется по алгоритму с последующей интеграцией информации об исправности всех модулей и исправности межмодульных связей в каждом узле для формирования интегральной исправности БЦВС в целом.

Тестирование БЦВС при изготовлении производится по следующему алгоритму. Узел формирования интегральной исправности расположен в модуле массовой памяти. Тест начального включения каждого модуля производит проверку исправности цифровой части модуля, а именно: ОЗУ, ПЗУ, процессорного элемента, ОЗУ устройств ввода-вывода (для модулей ввода-вывода) и внутримодульных каналов SpaceWire связи (для вычислительных модулей). Используется однозадачный режим для выполнения программы тестирования. Тест фонового контроля модуля во время загрузки ПО производит проверку исправности доступных для контроля ячеек ОЗУ, ПЗУ; процессорного элемента; ОЗУ устройств ввода-вывода (для модулей ввода-вывода); контрольных каналов устройств ввода-вывода (для модулей вводавывода) и внутримодульных каналов SpaceWire связи (для вычислительных модулей). В данном режиме выполняется тестовая программа на фоне выполнения программы загрузки в многозадачном режиме функционирования БЦВС. Тест встроенного контроля модуля производит проверку исправности доступных для контроля ячеек памяти ОЗУ, процессорного элемента, доступных для проверки ячеек ОЗУ устройств ввода-вывода (для интерфейсных модулей). Результат исправности передается по внутримодульным каналам SpaceWire (для вычислительных модулей). В данном режиме выполняется тестовая программа на фоне выполнения функциональной программы под управлением операционной системы в режиме разделения времени.

В состав ПО средств контроля исправности БЦВС класса ИМА входят следующие программные компоненты (ПК) и ПМ: 1) программный компонент управления режимами ПК-УР, обеспечивающий первичную инициализацию

(установку) регистров микропроцессора и управляющих регистров программируемых логических схем КФМ; проверку работоспособности КФМ по начальному включению (после первичной подачи питания, после рестарта БЦВС из-за кратковременного перерыва питания на объекте или из-за возникшей в процессе работы ошибки); анализ режима работы БЦВС (в составе объекта, при проверке на заводе-изготовителе в составе рабочего места); анализ режима работы тестового ПО: тест наземного контроля (ТНК), тест встроенного контроля (ТВК), тест фонового контроля; 2) программный компонент сервисного обслуживания ПК-СО - обеспечивает обмен данными по технологическим интерфейсам БЦВС с внешними устройствами (инструментальной ЭВМ, платами-имитаторами управляющих сигналов и т. д.) в режиме «Монитор», доступ к внутренним вычислительным ресурсам КФМ, загрузку и отладку тестового и функционального ПО; 3) программный компонент автономного контроля ПК-АК - обеспечивает процесс автоматизированного автономного контроля КФМ при наличии управляющего сигнала извне (на внешнем соединителе). Контроль исправности КФМ осуществляется без участия операционной системы и ФПО. В случае положительного результата проверки в фиксированных ячейках ОЗУ формируется информация, подтверждающая исправность КФМ, в случае отрицательного результата — информация об обнаруженных неисправностях. Компонент ПК-АК может выполняться из системного ПЗУ КФМ или может быть загружаемым извне и выполняться из ОЗУ КФМ.; 4) программный компонент исправности ПК-И - обеспечивает анализ состояния работоспособности КФМ, входящих в БЦВС, путем их опроса по межмодульному внутреннему интерфейсу SpaceWire и формирование интегральной исправности БЦВС в режиме автономного контроля; 5) библиотека программных компонентов штатного контроля БПК-ШК - обеспечивает проверку работоспособности КФМ в штатном режиме работы БЦВС. Каждый программный компонент в составе БПК-ШК представляет собой законченную программно реализуемую процедуру проверки, обеспечивающую контроль исправности функционального узла КФМ и работающую под управлением операционной системы или ФПО. Формирование интегральной исправности БЦВС осуществляется отдельной программной процедурой, также входящей в состав БПК-ШК; 6) программный компонент тестов наземного контроля ПК-ТНК - обеспечивает проверку БЦВС на заводе-изготовителе. Компонент входит в состав ПО АРМ (инструментальной ЭВМ) по проверке и настройке БЦВС и загружается в ОЗУ КФМ на время проверки в составе АРМ; 7) программный компонент обслуживания запроса ПК-ОЗ -обеспечивает обслуживание запроса о состоянии исправности КФМ по межмодульному внутрисистемному интерфейсу SpaceWire; 8) Программный компонент загрузки ПК-З - обеспечивает загрузку ФПО из МПП в оперативную память всех КФМ и его идентификацию по

контрольным признакам (контрольные суммы данных и программ, принадлежность ПК и ПМ к конкретному виду КФМ и др.).

Инструментальная программа проверки БЦВС обеспечивает занесение в БЦВС тестового ПО по технологическому каналу SpaceWire и обмен информацией по рабочим каналам SpaceWire с модулями БЦВС в режиме проверки. Инструментальная программа проверки имеет опции «Загрузка и поллинг» и «Таблица конфигурации» и представляет собой САПР, предназначенную для контроля БЦВС.

Взаимодействие БЦВС и программы контроля БЦВС, работающей на инструментальной ЭВМ, осуществляется через коммутаторы вычислительных модулей по сетевому интерфейсу SpaceWire с использованием системы логической адресации КФМ. Информация о результатах выполнения всех этапов проверки (загрузки и тестирования) отображается на экране инструментальной ЭВМ и в файлах отчета для каждого КФМ.

Изменения в принципах аппаратной и программной реализации изделий авионики, введенные в стандартах группы ARINC 651–ARINC 655, в значительной мере влияют на организацию процессов тестирования бортовых систем в целом. Специфическими требованиями, присущими рабочим местам по проверке интегрированной авионики, являются: повышенный уровень контроля аппаратурной составляющей изделий; возможность имитации состояния отказа отдельных компонентов авионики для проверки режима реконфигурирования вычислительной системы; модульное построение ПО с разделением тестов проверки на компоненты, исполняемые на уровне каждого КФМ и вычислителя в целом в однозадачном и многозадачном режимах; открытость архитектуры рабочего места, обеспечивающая возможность изменения уровня сложности контроля изделия и контроль изделий одного класса сложности; внутрипроектная унификация как аппаратных средств, так и ПО АРМ проверки.

Контрольные вопросы

1. Что понимается под термином диагностика вычислительных систем?
2. Перечислите показатели качества систем диагностики.
3. Поясните принципы организации процессов диагностирования.
4. Назначение диагностической модели.
5. Опишите диагностические модели ВС.
6. Поясните, как осуществляется формализованное задание вариантов ДС?
7. Назовите задачи, реализуемые в процессе отладки АСУ ТП.
8. Поясните схемы иерархических структур системы диагностирования.
9. Поясните алгоритм тестирование БЦВС.

Список используемой литературы

1. Межгосударственный стандарт Гост 27.002 -2015. Надежность в технике

Термины и определения //Эл. ресурс URL: https://files.stroyinf.ru/ Data2/1/4293754 /4293754027.pdf (дата размещения – постоянно).

2. ГОСТ 20911–89. Техническая диагностика. Термины и определения //Эл. ресурс URL:http://docs.cntd.ru/document/gost-20911-89(дата размещения – постоянно).

3. Байда, Н. П. Микропроцессорные системы поэлементного диагностирования РЭА / Н. П. Байда, И. В. Кузьмин, В. Т. Шпилевой. – М.: Радио и связь, 1987. – 256 с.

4. Барзилович Е. Ю. Оптимизация периодичности контроля систем, недоступных непрерывным проверкам, Автомат. и телемех., 1969, выпуск 8, с. 175–177 //Эл. ресурс URL: http://www.mathnet.ru/links/1d14f58e665d09a53d2dd a7fd86e038e/at10350.pdf (дата размещения – постоянно).

5. Заренин Ю.Г., Роик М.Е. Оценка и прогнозирование надежности программных средств.-М.: Машиностроение, 1986,-71 с.

6. Захарова О.Л., Кирсанова Ю.А., Книга Е.В., и др. Алгоритмы и программные средства тестирования бортовых цифровых вычислительных систем интегрированной модульной авионики /Информационно-управляющие системы, 2013г,№3-С.19-29 //Эл. ресурс URL: https://cyberleninka.ru/article/n /algoritmy-i-programmnye-sredstva-testirovaniya-bortovyh-tsifrovyh-vychislitelnyh-sistem-integrirovannoy-modulnoy-avioniki/viewer (дата размещения – постоянно).

7. Конесев С.Г., Хазиева Р.Т. Методы оценки показателей надежности сложных компонентов и систем // Современные проблемы науки и образования. –2015.–№1-1 //Эл. ресурс URL: http://www.science-education.ru/ru/article/view?id=17558 (дата обращения: 22.03.2020).

8. Липаев В. В. Тестирование компонентов и комплексов программ - Directmedia, 2015 – 528с. //Эл. ресурс URL: https://books.google.ru/booksid= PJk5CwAAQBAJ (дата размещения – постоянно).

9. Общие сведения по эксплуатация средств вычислительной техники Источник: http://refleader.ru/jgeotrbew.html (дата обращения 12.12.2019).

10. Михеев В.А. Системный анализ методов обеспечения и повышения надежности многофункциональной информационной системы //Эл.ресурс URL: https://cyberleninka.ru/article/n/sistemnyy-analiz-metodov-obespecheniya-i-povysheniya-nadezhnosti-mnogofunktsionalnoy-informatsionnoy-sistemy (дата обращения 02.12.2019).

11. Василенко Н.В., Макаров В.А. Модели оценки надежности программного обеспечения //Эл.ресурс URL:https://cyberleninka.ru/article/n/ modeli-otsenki-nadezhnosti-programmnogo-obespecheniya (дата размещения - постоянно).

12. Сафронов И.В. Способ повышения надежности изделий /Патент № 2605046 // Эл. ресурс URL: https://findpatent.ru/patent/242/2424572.html 1.6.5 (дата обращения 21.01.2020).

13. Верещагин И. разработка ПО авионики // Эл. ресурс URL: https://findpatent.ru/patent/242/2424572.html 1.6.5 (дата обращения 20.11.19).