ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р МЭК 61508-6–2012

ФУНКЦИОНАЛЬНАЯ БЕЗОПАСНОСТЬ СИСТЕМ ЭЛЕКТРИЧЕСКИХ, ЭЛЕКТРОННЫХ, ПРОГРАММИРУЕМЫХ ЭЛЕКТРОННЫХ, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ

Часть 6

Руководство по применению ГОСТ Р МЭК 61508-2 и ГОСТ Р МЭК 61508-3

IEC 61508-6:2010

Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3 (IDT)

Издание официальное

Москва
Стандартинформ
2014

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации – ГОСТ Р 1.0–2004 «Стандартизации в Российской Федерации. Основные положения»

Сведения о стандарте

1 ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Корпоративные электронные системы» и Федеральным бюджетным учреждением «Консультационно-внедренческая фирма в области международной стандартизации и сертификации - «Фирма «ИНТЕРСТАНДАРТ» на основе собственного аутентичного перевода стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 58 «Функциональная безопасность»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии России от 29 октября 2012 г. № 591-ст

Настоящий стандарт идентичен международному стандарту МЭК 61508-6:2010 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 6. Руководство по применению МЭК 61508-2 и МЭК 61508-3» (IEC 61508-5:2010 «Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6. Guidelines on the application of IEC 61508-2 and IEC 61508-3»).

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5 (подраздел 3.5).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты, сведения о которых приведены в дополнительном приложении ДА

1 ВВЕДЕН взамен ГОСТ Р МЭК 61508-6-2007

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок – в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования – на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2013

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения .....
2 Нормативные ссылки .....
3 Термины и определения .....
Приложение А (справочное) Применение МЭК 61508-2 и МЭК 61508-3.....
Приложение В (справочное) Метод оценки вероятностей отказа аппаратных средств .....
Приложение С (справочное) Расчет охвата диагностикой и доли безопасных отказов.....
Приложение D (справочное) Методика количественного определения влияния отказов аппаратных средств по общей причине в Э/Э/ПЭ системах .....
Приложение Е (справочное) Применение таблиц полноты безопасности программного обеспечения в соответствии с МЭК 61508-3 .....
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации.....
Библиография .....

Введение

Системы, состоящие из электрических и/или электронных элементов, в течение многих лет используются для выполнения функций безопасности в большинстве областей применения. Компьютерные системы (обычно называемые «программируемые электронные системы»), применяемые во всех прикладных отраслях для выполнения функций, не связанных с безопасностью, во все более увеличивающихся количествах используются для выполнения функций обеспечения безопасности. Для эффективной и безопасной эксплуатации технологий, основанных на использовании компьютерных систем, чрезвычайно важно, чтобы лица, ответственные за принятие решений, имели в своем распоряжении руководства по вопросам безопасности, которые они могли бы использовать в своей работе.

Настоящий стандарт устанавливает общий подход к вопросам обеспечения безопасности для всего жизненного цикла систем, состоящих из электрических и/или электронных, и/или программируемых электронных (Э/Э/ПЭ) элементов, которые используются для выполнения функций обеспечения безопасности. Этот унифицированный подход был принят для разработки рациональной и последовательной технической политики для всех электрических систем обеспечения безопасности. При этом основной целью является содействие разработке стандартов для продукции и областей применения на основе стандартов серии МЭК 61508.

Примечание – Примерами стандартов для продукции и областей применения, разработанных на основе стандартов серии МЭК 61508, являются [1] – [3].

Обычно безопасность достигается за счет использования нескольких систем, в которых используются различные технологии (например, механические, гидравлические, пневматические, электрические, электронные, программируемые электронные). Любая стратегия безопасности должна, следовательно, учитывать не только все элементы, входящие в состав отдельных систем (например, датчики, управляющие устройства и исполнительные механизмы), но также и все подсистемы безопасности, входящие в состав общей системы обеспечения безопасности. Таким образом, хотя настоящий стандарт посвящен в основном Э/Э/ПЭ системам, связанным с безопасностью, он может также предоставлять общий подход, в рамках которого рассматриваются системы, связанные с безопасностью, базирующиеся на других технологиях.

Признанным фактом является существование огромного разнообразия использования Э/Э/ПЭ систем в различных областях применения, отличающихся различной степенью сложности, возможными рисками и опасностями. В каждом

конкретном применении необходимые меры безопасности будут зависеть от многочисленных факторов, специфичных для конкретного применения. Настоящий стандарт, являясь базовым, позволит формулировать такие меры для областей применения будущих международных стандартов, а также для последующих редакций уже существующих стандартов.

Настоящий стандарт:

- рассматривает все соответствующие стадии жизненного цикла безопасности систем в целом, а также подсистем Э/Э/ПЭ системы и программного обеспечения (например, от первоначальной концепции, через проектирование, внедрение, эксплуатацию и техническое обеспечение до снятия с эксплуатации), в ходе которых Э/Э/ПЭ системы используются для выполнения функций безопасности;

- был задуман с учетом быстрого развития технологий; его основа является в значительной мере устойчивой и полной для будущих разработок;

- делает возможной разработку стандартов областей применения, в которых используются Э/Э/ПЭ системы, связанные с безопасностью; разработка стандартов для областей применения в рамках общей структуры, вводимой настоящим стандартом, должна привести к более высокому уровню согласованности (например, основных принципов, терминологии, и т.д.) как для отдельных областей применения, так и для их совокупностей, что даст преимущества в плане безопасности и экономики;

- предоставляет метод разработки спецификации требований к безопасности, необходимых для достижения заданной функциональной безопасности Э/Э/ПЭ систем, связанных с безопасностью;

- использует для определения требований к уровням полноты безопасности подход, основанный на оценке рисков;

- вводит уровни полноты безопасности для определения целевого уровня полноты безопасности для функций безопасности, которые должны быть реализованы Э/Э/ПЭ системами, связанными с безопасностью.

Примечание – Настоящий стандарт не устанавливает требований к уровню полноты безопасности для любой функции безопасности, и не определяет, как устанавливается уровень полноты безопасности. Однако настоящий стандарт формирует основанный на риске концептуальный подход и предлагает примеры методов;

– устанавливает целевые величины отказов для функций безопасности, реализуемых Э/Э/ПЭ системами, связанными с безопасностью, и связывает эти меры с уровнями полноты безопасности;

– устанавливает нижнюю границу для целевых мер отказов для функции безопасности, реализуемой одиночной Э/Э/ПЭ системой, связанной с безопасностью. Для Э/Э/ПЭ систем, связанных с безопасностью в режиме:

- низкой интенсивности запросов на обслуживание: нижняя граница для выполнения функции, для которой система предназначена, устанавливается в соответствии со средней вероятностью опасного отказа по запросу, равной 10-5,
- высокой интенсивности запросов на обслуживание или в непрерывном режиме: нижняя граница устанавливается в соответствии со средней частотой опасных отказов 10-9 в час.

Примечания

1 Одиночная Э/Э/ПЭ система, связанная с безопасностью, не обязательно предполагает одноканальную архитектуру.

2 В проектах систем, связанных с безопасностью и имеющих низкий уровень сложности, можно достигнуть более низких значений целевой полноты безопасности, но предполагается, что в настоящее время указанные предельные значения целевой полноты безопасности могут быть достигнуты для относительно сложных систем (например, программируемые электронные системы, связанные с безопасностью);

– устанавливает требования по предотвращению и управлению систематическими отказами, основанные на опыте и заключениях из практического опыта. Учитывая, что вероятность возникновения систематических отказов, в общем случае, не может быть определена количественно, настоящий стандарт позволяет утверждать для специфицируемой функции безопасности, что целевая мера отказов, связанных с этой функцией, может считаться достигнутой, если все требования стандарта были выполнены.

– вводит понятие «стойкость к систематическим отказам», применяемое к элементу, характеризующее уверенность в том, что полнота безопасности, касающаяся систематических отказов элемента, удовлетворяет требованиям заданного уровня полноты безопасности;

– применяет широкий диапазон принципов, методов и средств для достижения функциональной безопасности Э/Э/ПЭ систем, связанных с безопасностью, но не использует явно понятие «безопасного отказа». В то же время, понятия «безопасный отказ» и «безопасный в своей основе» могут быть использованы, но для этого необходимо обеспечить соответствующие требования в конкретных разделах стандарта, которым эти понятия должны соответствовать.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФУНКЦИОНАЛЬНАЯ БЕЗОПАСНОСТЬ СИСТЕМ ЭЛЕКТРИЧЕСКИХ, ЭЛЕКТРОННЫХ, ПРОГРАММИРУЕМЫХ ЭЛЕКТРОННЫХ, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ

Часть 6.

Руководство по применению ГОСТ Р МЭК 61508-2 и ГОСТ Р МЭК 61508-3

Electric systems. Functional safety of electrical/electronic/programmable electronic safety-related systems.
Part 6. Guidelines on the application of IEC 61508-2 and IEC 61508-3

Дата введения – 2013 – 08 – 01

1 Область применения

1.1 Настоящий стандарт содержит информацию и руководящие указания по применению МЭК 61508-2 и МЭК 61508-3.

Краткий обзор требований МЭК 61508-2 и МЭК 61508-3 и определение функциональной последовательности их применения содержится в приложении А.

Пример методики расчета вероятности отказа аппаратных средств содержится в приложении В, которое следует применять совместно с МЭК 61508-2 (п. 7.4.3 и приложение С) и приложением D настоящего стандарта.

Пример расчета охвата диагностикой содержится в приложении С, которое следует применять совместно с МЭК 61508-2 (приложение С).

Метод количественной оценки влияния отказов аппаратных средств по общей причине на вероятность отказов описан в приложении D.

Примеры применения таблиц полноты безопасности программного обеспечения, приведенных в МЭК 61508-3 (приложение А), для уровней полноты безопасности 2 и 3 описаны в приложении Е.

1.2 МЭК 61508-1, МЭК 61508-2, МЭК 61508-3 и МЭК 61508-4 являются базовыми стандартами по безопасности, хотя этот статус не применим в контексте Э/Э/ПЭ систем, связанных с безопасностью, имеющих низкую сложность (см. МЭК 61508-4, пункт 3.4.3). В качестве базовых стандартов по безопасности данные стандарты предназначены для использования техническими комитетами при подготовке стандартов в соответствии с принципами, изложенными в руководстве МЭК 104 руководстве ИСО/МЭК 51. Следующие стандарты, МЭК 61508-1, МЭК 61508-2, МЭК 61508-3 и МЭК 61508-4 предназначены для использования в качестве самостоятельных стандартов. Функция

Издание официальное

безопасности настоящего стандарта не применима к медицинскому оборудованию, удовлетворяющему требованиям серии горизонтальных стандартов МЭК 60601 [1]..

1.3 В круг обязанностей Технического комитета входит использование там, где это возможно, основополагающих стандартов по безопасности при подготовке собственных стандартов. В этом случае требования, методы проверки или условия проверки настоящего основополагающего стандарта по безопасности не будут применяться, если на них нет конкретной ссылки, или они не включены в публикации, подготовленные этими техническими комитетами.

1.4 Общая структура стандартов серии МЭК 61508 и роль, которую играет настоящий стандарт в достижении функциональной безопасности Э/Э/ПЭ систем, связанных с безопасностью, показана на рисунке 1.

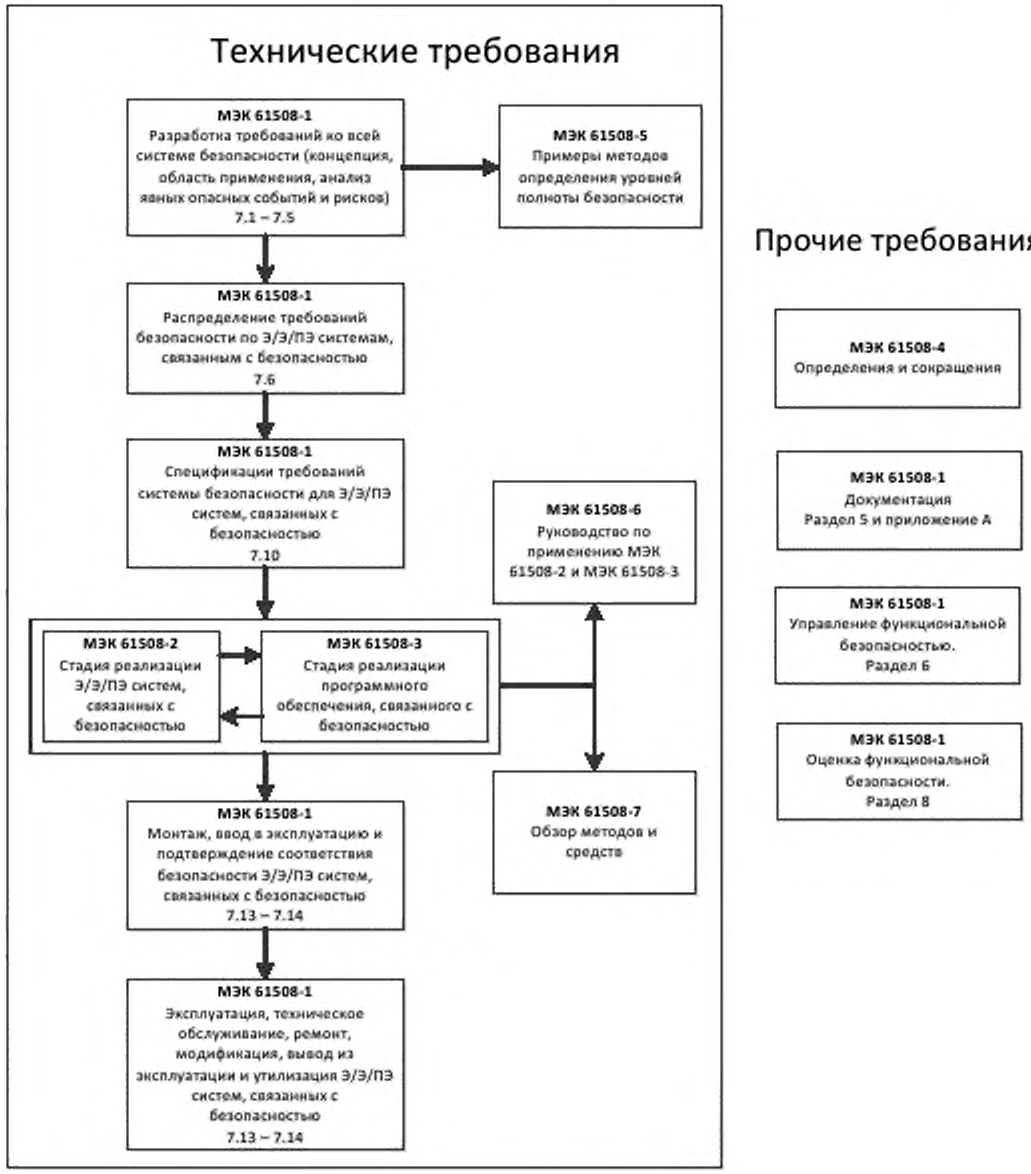


Рисунок 1 – Общая структура серии ГОСТ Р МЭК 61508

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ИСО/МЭК Руководство 51:1990 Аспекты безопасности. Руководящие указания по включению в стандарты (ISO/IEC Guide 51:1999, Safety aspects – Guidelines for their inclusion in standards)

МЭК Руководство 104:1997 Подготовка публикаций по безопасности и использование базовых публикаций по безопасности и публикаций по безопасности групп (IEC Guide 104:1997, The preparation of safety publications and the use of basic safety publications and group safety publications)

МЭК 61508-1:2010 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 1. Общие требования (IEC 61508-1:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements)

МЭК 61508-2:2010 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 2. Требования к системам электрическим/электронным/программируемым электронным, связанным с безопасностью (IEC 61508-4:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2. Requirements for electrical / electronic / programmable electronic safety-related systems)

МЭК 61508-3:2010 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 3. Требования к программному обеспечению (IEC 61508-3:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements)

МЭК 61508-4:2010 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 4. Термины и определения (ISO/IEC 61508-4:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4. Definitions and abbreviations)

МЭК 61508-5:2010 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 5. Примеры методов определения уровней полноты безопасности (IEC 61508-5:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 5. Examples of methods for the determination of safety integrity levels)

МЭК 61508-7:2010 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 7.

Анализ методов и средств (IEC 61508-7:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures)

3 Термины, определения и сокращения

В настоящем стандарте используются термины, определения и сокращения по МЭК 61508-4.

Приложение А
(справочное)

Применение МЭК 61508-2 и МЭК 61508-3

А.1 Общие положения

Конкретный механизм, технологическая установка, а также другое оборудование могут в случае неправильной работы (например отказ электрических, электронных и/или программируемых электронных устройств) представлять опасность для людей и окружающей среды из-за возникновения опасных событий (например пожары, взрывы, избыточная радиация, попадание в механизмы и т.д.). Аварии оборудования могут возникать по причине физических отказов устройств (неожиданные аварии оборудования), либо систематических отказов (ошибки человека в технических условиях и конструкции конкретной системы при определенной комбинации причин приводят к систематическим отказам), либо некоторых внешних условий.

Общий подход, основанный на оценке рисков, для предотвращения и/или управления отказами в электромеханических, электронных или программируемых электронных устройствах содержится в МЭК 61508-1.

Основная задача настоящего стандарта заключается в том, чтобы установки и оборудование были обеспечены автоматизированными системами безопасности, а его основная цель состоит в предотвращении:

- отказов систем управления, инициирующих другие события, которые, в свою очередь, могут привести к опасному событию (например утечка токсичных материалов, повторяющиеся удары механизмов и т.д.) и

- необнаруженных отказов систем защиты (например в системах аварийной остановки), делающих эти системы недоступными в момент необходимости действий, связанных с безопасностью.

Требование проведения анализа опасности и риска для процесса/механизма для определения степени снижения риска, необходимой для удовлетворения критериям оценки риска для данного применения, см. в МЭК 61508-1. Оценка риска основана на оценке как последствий (или серьезности), так и частоты (или вероятности) опасного события.

Требование использования степени снижения риска, определенной в процессе анализа, для решения вопроса о том, требуется одна или несколько систем, связанных с

безопасностью[1)], и для выполнения каких функций обеспечения безопасности (каждая с заданной полнотой безопасности[2)]) требуются эти системы, содержится в МЭК 61508-1.

В МЭК 61508-2 и МЭК 61508-3 рассматриваются требования к функциям безопасности и полноте безопасности, установленные в МЭК 61508-1, для любой Э/Э/ПЭ системы, связанной с безопасностью, а также устанавливаются требования к жизненному циклу системы безопасности, которые:

- применяются при разработке технического задания, проектировании и внесении изменений в аппаратные средства и программное обеспечение, а также
- фокусируются на средствах предотвращения и/или управления случайными отказами аппаратных средств и систематическими отказами (жизненные циклы Э/Э/ПЭ системы безопасности и программного обеспечения[3)] системы безопасности).

МЭК 61508-2 и МЭК 61508-3 не содержат указаний, какой уровень полноты безопасности соответствует заданному требуемому приемлемому риску. Это решение зависит от многих факторов, включая характер применения, степень выполнения функций безопасности другими системами, а также социальные и экономические факторы (см. МЭК 61508-1 и МЭК 61508-5).

Требования МЭК 61508-2 и МЭК 61508-3 включают в себя:

- применение методов[4)] и средств, классифицированных в соответствии с

[1)] Системы, необходимые для обеспечения функциональной безопасности и содержащие одно или несколько электрических (электромеханических), электронных или программируемых электронных (Э/Э/ПЭ) устройств, называются системами Э/Э/ПЭ, связанными с безопасностью, и включают в себя все оборудование, необходимое для реализации требуемой функции безопасности (см. МЭК 61508-4, п. 3.5.1).

[2)] Уровень полноты безопасности определяется как один из четырех дискретных уровней. Уровень полноты безопасности 4 является наивысшим, а уровень полноты безопасности 1 – низшим (см. МЭК 61508-4, пункты 3.5.4 и 3.5.8).

[3)] Чтобы сделать возможной четкую структуризацию требований настоящего стандарта, было принято решение упорядочить требования с помощью модели процесса разработки, в которой все этапы следуют в четкой последовательности с небольшим шагом (ее иногда называют потоковой моделью). Однако следует подчеркнуть, что может быть использован любой подход к описанию жизненного цикла при условии, что он будет эквивалентен описанному в плане проектирования системы безопасности (см. МЭК 61508-1, раздел 7).

[4)] Требуемые методы и средства для каждого уровня полноты безопасности представлены в МЭК 61508-2 (таблицы приложений А и В) и МЭК 61508-3.

уровнем полноты безопасности, чтобы избежать систематических отказов[1)] с помощью планово-предупредительных мер, и

- управление систематическими отказами (включая отказы программного обеспечения) и случайными отказами аппаратных средств с помощью конструктивных особенностей, таких как встроенные средства обнаружения повреждений, избыточность и особенности архитектуры (например диверсификация).

В МЭК 61508-2 гарантия того, что нужный уровень полноты безопасности будет удовлетворительным для опасных случайных отказов аппаратных средств, основывается на:

- требованиях к отказоустойчивости аппаратуры (см. МЭК 61508-2, таблицы 2 и 3) и

- диагностическом охвате и частоте контрольных испытаний подсистем и компонентов с проведением анализа надежности, использующего соответствующие данные.

В МЭК 61508-2 и МЭК 61508-3 гарантия того, что нужный уровень полноты безопасности будет удовлетворительным для систематических отказов, достигается путем:

- правильного применения процедур управления безопасностью;
- использования компетентного персонала;
- выполнения предусмотренных действий по реализации жизненного цикла системы безопасности, включая предусмотренные методы и средства[2)] и
- выполнения независимой оценки функциональной безопасности[3)].

Главная цель состоит в обеспечении того, что оставшиеся систематические отказы, соответствующие уровню полноты безопасности, не приведут к отказу Э/Э/ПЭ системы, связанной с безопасностью.

[1)] Систематические отказы обычно нельзя определить количественно. Причинами отказов бывают: ошибки при спецификации и проектировании технических средств и программного обеспечения; ошибки при учете условий окружающей среды (например температуры) и ошибки в процессе работы (например слабый интерфейс).

[2)] Средства, альтернативные описанным в настоящем стандарте, можно использовать при условии, что при планировании обеспечения безопасности документально оформляется обоснование использования альтернативных средств (см. МЭК 61508-1, раздел 6).

[3)] Независимая оценка не всегда подразумевает проведение оценки третьей стороной (см. МЭК 61508-1, раздел 8).

МЭК 61508-2 был разработан, чтобы формализовать требования к обеспечению полноты безопасности аппаратных средств[1)] Э/Э/ПЭ систем, связанных с безопасностью, включая датчики и исполнительные элементы. Необходимы методы и средства, направленные против как случайных, так и систематических отказов аппаратных средств. Они, как указано выше, включают в себя соответствующую комбинацию средств по предотвращению неисправностей и управлению отказами. Если для обеспечения функциональной безопасности необходимы действия оператора, то приводятся требования к интерфейсу оператора. В МЭК 61508-2 для обнаружения случайных отказов аппаратных средств также определяются методы и средства диагностического тестирования, реализуемые на уровне программного обеспечения и аппаратных средств (например, диверсификация).

МЭК 61508-3 был разработан, чтобы формализовать требования обеспечения полноты безопасности для программного обеспечения как встроенного (включая диагностические средства обнаружения неисправностей), так и прикладного. МЭК 61508-3 требует использовать комбинированный подход, включающий исключение ошибок (обеспечение качества) и устойчивость к ошибкам (за счет архитектуры программного обеспечения), так как не существует известного способа проверить отсутствие отказов в достаточно сложном программном обеспечении, связанном с безопасностью, и, особенно, избежать ошибок в технических условиях и в проекте. МЭК 61508-3 требует принятия таких принципов разработки программного обеспечения, как проектирование сверху вниз, модульность, проверка на каждой стадии жизненного цикла разработки, проверка программных модулей и библиотек программных модулей, а также четкое документирование для облегчения проверки и подтверждения соответствия. Для различных уровней программного обеспечения требуются различные уровни гарантии того, что эти и связанные с ними принципы были правильно реализованы.

Разработчик программного обеспечения может быть или не быть частью организации, создающей всю Э/Э/ПЭ систему. В любом случае необходимо тесное сотрудничество, особенно при разработке архитектуры программируемой электроники, когда требуется анализировать компромиссы между архитектурами аппаратных средств и программного обеспечения при оценке их вклада в обеспечение безопасности (см.

[1)] Включая постоянное встроенное программное обеспечение или эквиваленты программного обеспечения (также называемые программно-аппаратными средствами), например специализированные для применения интегральные схемы.

МЭК 61508-2, рисунок 4).

А.2 Функциональные этапы применения МЭК 61508-2

Функциональные этапы применения МЭК 61508-2 представлены в настоящем приложении на рисунках А.1 и А.2. Функциональные этапы применения МЭК 61508-3 представлены на рисунке А.3.

Для МЭК 61508-2 можно выделить следующие функциональные этапы (см. рисунки А.1 и А.2):

a) Определяют распределение требований к системе безопасности (МЭК 61508-1). При необходимости выполняют обновление планирования подтверждения соответствия системе безопасности в процессе разработки Э/Э/ПЭ системы, связанной с безопасностью.

b) Определяют требования к Э/Э/ПЭ системам, связанным с безопасностью, включая требования к полноте безопасности для каждой функции безопасности (МЭК 61508-2, подраздел 7.2). Определяют требования к программному обеспечению и передают их поставщику и/или разработчику программного обеспечения для применения МЭК 61508-3.

Примечание – На этой стадии необходимо рассмотреть возможность одновременных отказов в системе управления УО и Э/Э/ПЭ системе (системах), связанной с безопасностью, (см. МЭК 61508-5, А.5.4). Такие отказы могут быть результатом отказов компонентов по общей причине, например, из-за влияния окружающей среды. Наличие подобных отказов может привести к большим, по сравнению с ожидаемым, значениям остаточного риска.

c) Начинают планирование подтверждения соответствия безопасности для Э/Э/ПЭ системы (см. МЭК 61508-2, подраздел 7.3).

d) Задают архитектуру (конфигурацию) логической подсистемы, датчиков и исполнительных элементов. Вместе с поставщиком/разработчиком программного обеспечения анализируют архитектуру аппаратных средств, программного обеспечения и влияние на безопасность компромиссов между аппаратными средствами и программным обеспечением (см. МЭК 61508-2, рисунок 4). При необходимости анализ повторяют.

e) Разрабатывают модель архитектуры аппаратных средств для Э/Э/ПЭ системы, связанной с безопасностью. Эту модель разрабатывают, проверяя отдельно каждую функцию безопасности, и определяют подсистему (компонент), используемую для реализации этой функции.

f) Устанавливают параметры для каждой подсистемы (компонента), используемой в Э/Э/ПЭ системе, связанной с безопасностью. Для каждой подсистемы (компонента) определяют:

- временной интервал между тестовыми испытаниями для отказов, которые не обнаруживаются автоматически;
- среднее время восстановления;
- диагностический охват (см. МЭК 61508-2, приложение С);
- вероятность отказа;
- долю безопасных отказов (см. МЭК 61508-2, приложение С).
- требуемые архитектурные ограничения; для Способа 1_H см. МЭК 61508-2, п. 7.4.4.2 и приложение С, а для Способа 2_H см. МЭК 61508-2, п. 7.4.4.3.

g) Создают модель расчета безотказности для каждой функции безопасности, которую должна реализовать Э/Э/ПЭ система, связанная с безопасностью.

Примечание – Модель расчета безотказности представляет собой математическую формулу, показывающую взаимосвязь между безотказностью и соответствующими параметрами, связанными с оборудованием и условиями его использования.

h) Рассчитывают прогнозируемую безотказность для каждой функции безопасности, используя соответствующую методику. Сравнивают результат с заданными характеристиками отказов, определенными в перечислении b), и требованиями для Способа 1_H (см. МЭК 61508-2, 7.4.4.2) или Способа 2_H (см. МЭК 61508-2, 7.4.4.3). Если прогнозируемая безотказность не соответствует заданным характеристикам отказов и/или требованиям Способа 1_H или Способа 2_H, то изменяют:

- если возможно, один или несколько параметров подсистемы [возвращаются к перечислению f)] и/или
- архитектуру аппаратных средств [возвращаются к перечислению d)].

Примечание – Существует множество методов моделирования, и аналитик должен выбрать наиболее соответствующий (перечень некоторых методов, которые могут быть использованы, приведен в приложении В).

i) Реализуют проект Э/Э/ПЭ системы, связанной с безопасностью. Выбирают средства и методы для управления систематическими отказами аппаратных средств, отказами, вызванными влиянием окружающей среды, и эксплуатационными отказами (см. МЭК 61508-2, приложение А).

j) Загружают проверенное программное обеспечение (см. МЭК 61508-3) в соответствующие аппаратные средства (см. МЭК 61508-2, подраздел 7.5 и приложение В) и параллельно разрабатывают рабочие инструкции для пользователей и документацию для обслуживающего персонала по технической эксплуатации системы (см. МЭК 61508-2, подраздел 7.6 и приложение В). Учитывают аспекты, связанные с программным обеспечением (см. А.3, перечисление f)).

k) Вместе с разработчиком программного обеспечения (см. МЭК 61508-3, подраздел 7.7) проводят подтверждение соответствия безопасности Э/Э/ПЭ системы (см. МЭК 61508-2, подраздел 7.7 и приложение В).

l) Передают аппаратные средства и результаты подтверждения соответствия Э/Э/ПЭ системы, связанной с безопасностью, системе безопасности системным инженерам для дальнейшей интеграции всей системы.

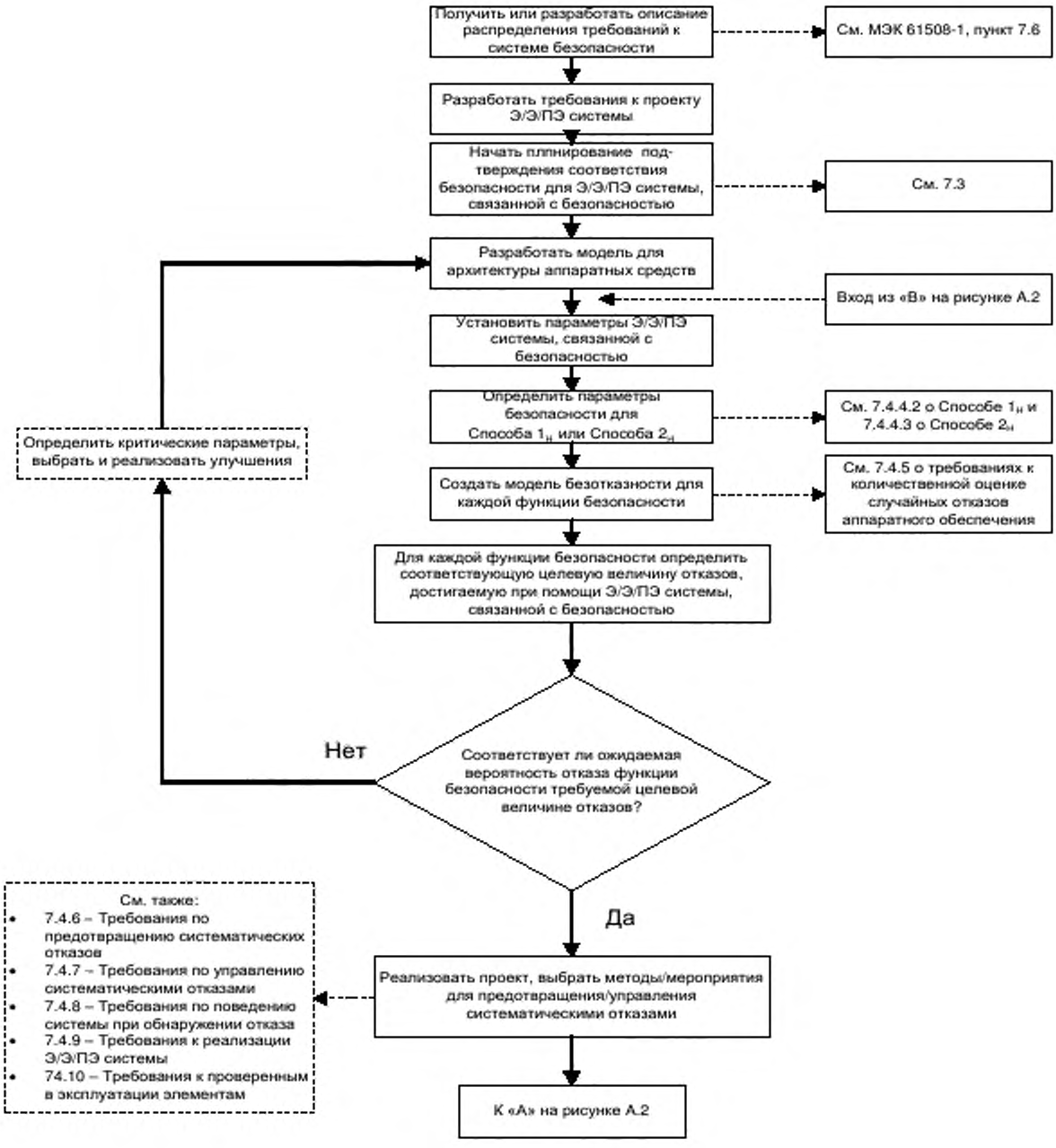
m) Если в процессе эксплуатации Э/Э/ПЭ системы, связанной с безопасностью, требуется модернизация/техническое обслуживание, то при необходимости снова обращаются к МЭК 61508-2, подраздел 7.8.

В процессе жизненного цикла системы безопасности для Э/Э/ПЭ системы, связанной с безопасностью, выполняется множество различных действий. Среди них верификация (см. МЭК 61508-2, подраздел 7.9) и оценка функциональной безопасности (см. МЭК 61508-1, раздел 8).

В процессе выполнения приведенных выше действий для Э/Э/ПЭ системы, связанной с безопаснотью, выбирают обеспечивающие безопасность методы и средства, соответствующие требуемому уровню полноты безопасности. Для помощи с выбором таких методов и средств составлены таблицы, упорядочивающие различные методы/средства в соответствии с четырьмя уровнями полноты безопасности (см. МЭК 61508-2, приложение В). Краткий обзор каждого из методов и средств со ссылками на источники информации о них, включая перекрестные ссылки на эти таблицы, представлен в МЭК 61508-7, приложения А и В.

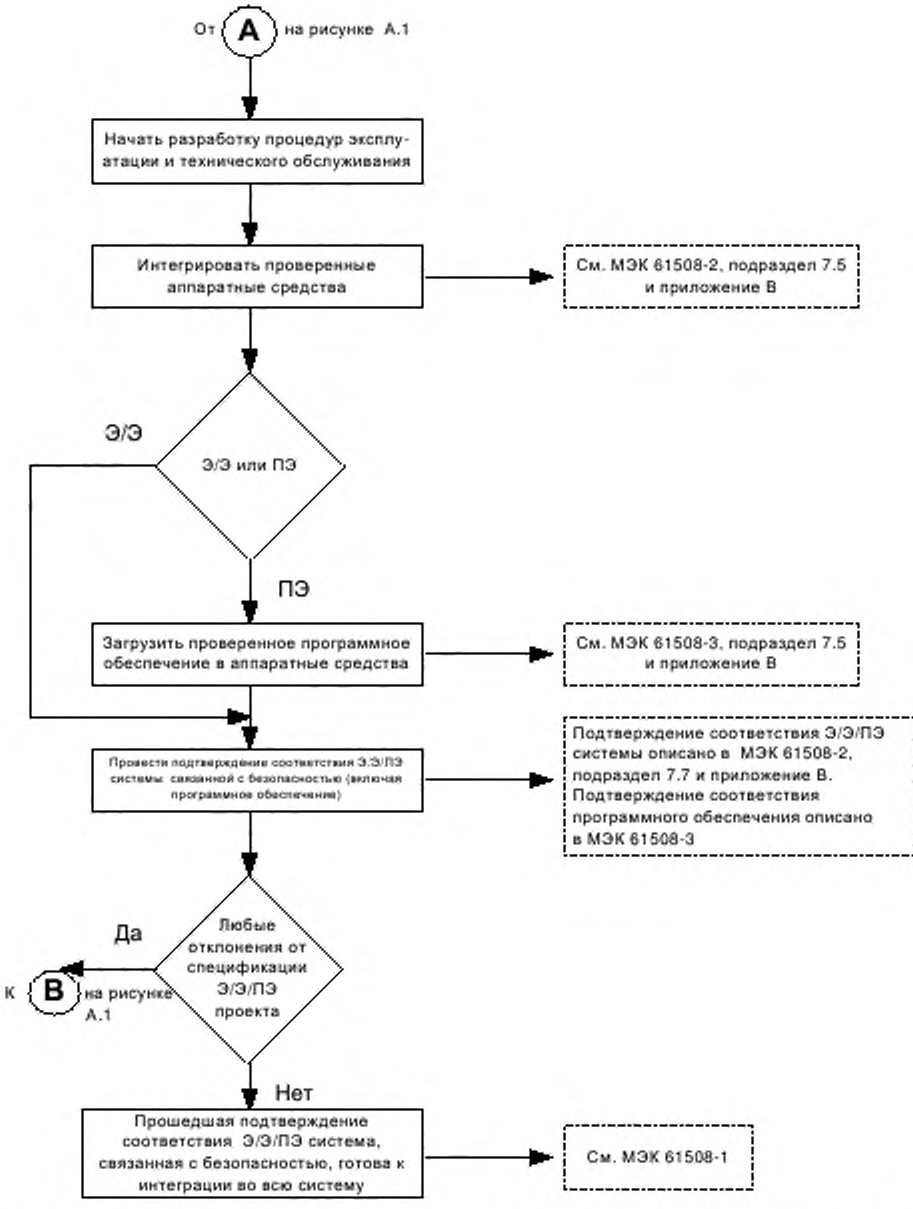
Один из возможных методов расчета вероятностей отказа аппаратных средств для Э/Э/ПЭ систем, связанных с безопасностью, представлен в приложении В.

Примечание – При выполнении приведенных выше действий допускается применять средства, альтернативные указанным в настоящем стандарте, при условии, что оправдывающие обстоятельства документально оформляются в процессе планирования подтверждения соответствия системе безопасности (см. МЭК 61508-1, раздел 6).



Примечание – В ПЭ системах, связанных с безопасностью, для программного обеспечения выполняются аналогичные действия (см. рисунок А.3).

Рисунок А.1 – Функциональные этапы применения МЭК 61508-2



Примечание – В ПЭ системах, связанных с безопасностью, для программного обеспечения выполняются аналогичные действия (см. рисунок А.3).

Рисунок А.2 – Функциональные этапы применения МЭК 61508-2 (продолжение)

А.3 Функциональные этапы применения МЭК 61508-3

Можно выделить следующие функциональные этапы применения МЭК 61508-3

(см. рисунок А.3):

a) Определяют требования для систем Э/Э/ПЭ, связанных с безопасностью, и соответствующих компонент планирования подтверждения соответствия системе безопасности (см. МЭК 61508-2, подраздел 7.3). При необходимости выполняют обновление планирования подтверждения соответствия системе безопасности в процессе разработки программного обеспечения.

Примечание – На предыдущих стадиях жизненного цикла были:

- определены требуемые функции безопасности и соответствующие им уровни полноты безопасности (см. МЭК 61508-1, подразделы 7.4 и 7.5);
- распределены функции безопасности для назначенных систем Э/Э/ПЭ, связанных с безопасностью (см. МЭК 61508-1, подраздел 7.6) и
- распределены реализуемые программно функции внутри каждой системы Э/Э/ПЭ, связанной с безопасностью (см. МЭК 61508-2, подраздел 7.2).

b) Определяют архитектуру программного обеспечения для всех реализуемых программно функций безопасности (см. МЭК 61508-3, подраздел 7.4 и приложение А).

c) Вместе с поставщиком/разработчиком Э/Э/ПЭ системы, связанной с безопасностью, анализируют архитектуру аппаратных средств и программного обеспечения и влияние на безопасность компромиссов между аппаратными средствами и программным обеспечением (см. МЭК 61508-2, рисунок 4). При необходимости анализ повторяют.

d) Приступают к планированию проверки и подтверждения соответствия безопасности для программного обеспечения (см. МЭК 61508-3, подразделы 7.3 и 7.9).

e) Проектируют, разрабатывают и проверяют/тестируют программное обеспечение в соответствии с:

- планированием подтверждения соответствия безопасности для программного обеспечения;
- уровнем полноты безопасности программного обеспечения;
- жизненным циклом программного обеспечения системы безопасности.

f) Завершают действия по окончательной проверке программного обеспечения и интегрируют проверенное программное обеспечение в соответствующие аппаратные средства (см. МЭК 61508-3, подраздел 7.5) и параллельно разрабатывают процедуры по аспектам программного обеспечения для пользователей и для обслуживающего персонала системы, выполняемые при эксплуатации системы (см. МЭК 61508-3, подраздел 7.6, а также приложение А, подраздел А.2, перечисление k) настоящего стандарта).

g) Вместе с разработчиком аппаратных средств (см. МЭК 61508-2, подраздел 7.7) проводят подтверждение соответствия программного обеспечения в интегрированных Э/Э/ПЭ системах, связанных с безопасностью (см. МЭК 61508-3, подраздел 7.7).

h) Передают результаты подтверждения соответствия безопасности Э/Э/ПЭ системы, связанной с безопасностью, системным инженерам для дальнейшей интеграции всей системы.

i) Если в процессе эксплуатации потребуется модернизация программного обеспечения Э/Э/ПЭ системы, связанной с безопасностью, то выполняется возврат к соответствующей стадии, как описано в МЭК 61508-3, подраздел 7.8.

В процессе жизненного цикла программного обеспечения системы безопасности выполняется множество различных действий. Среди них верификация (см. МЭК 61508-2, подраздел 7.9) и оценка функциональной безопасности (см. МЭК 61508-1, раздел 8).

В процессе выполнения приведенных выше действий выбирают обеспечивающие безопасность программного обеспечения методы и средства, соответствующие требуемому уровню полноты безопасности. Для помощи с выбором таких методов и средств составлены таблицы, упорядочивающие различные методы/средства в соответствии с четырьмя уровнями полноты безопасности (см. МЭК 61508-3, приложение А). Краткий обзор каждого из методов и средств со ссылками на источники информации о них, включая перекрестные ссылки на эти таблицы, представлен в МЭК 61508-7, приложение С.

Обработанные примеры применения таблиц полноты безопасности приведены в приложении Е настоящего стандарта, а МЭК 61508-7 включает в себя описание вероятностного подхода к определению полноты безопасности программного обеспечения для уже разработанного программного обеспечения (см. МЭК 61508-7, приложение D).

Примечание – При выполнении приведенных выше действий допускается применять средства, альтернативные указанным в настоящем стандарте, при условии, что соответствующее обоснование документально оформляется в процессе планирования подтверждения соответствия системе безопасности (см. МЭК 61508-1, раздел 6).

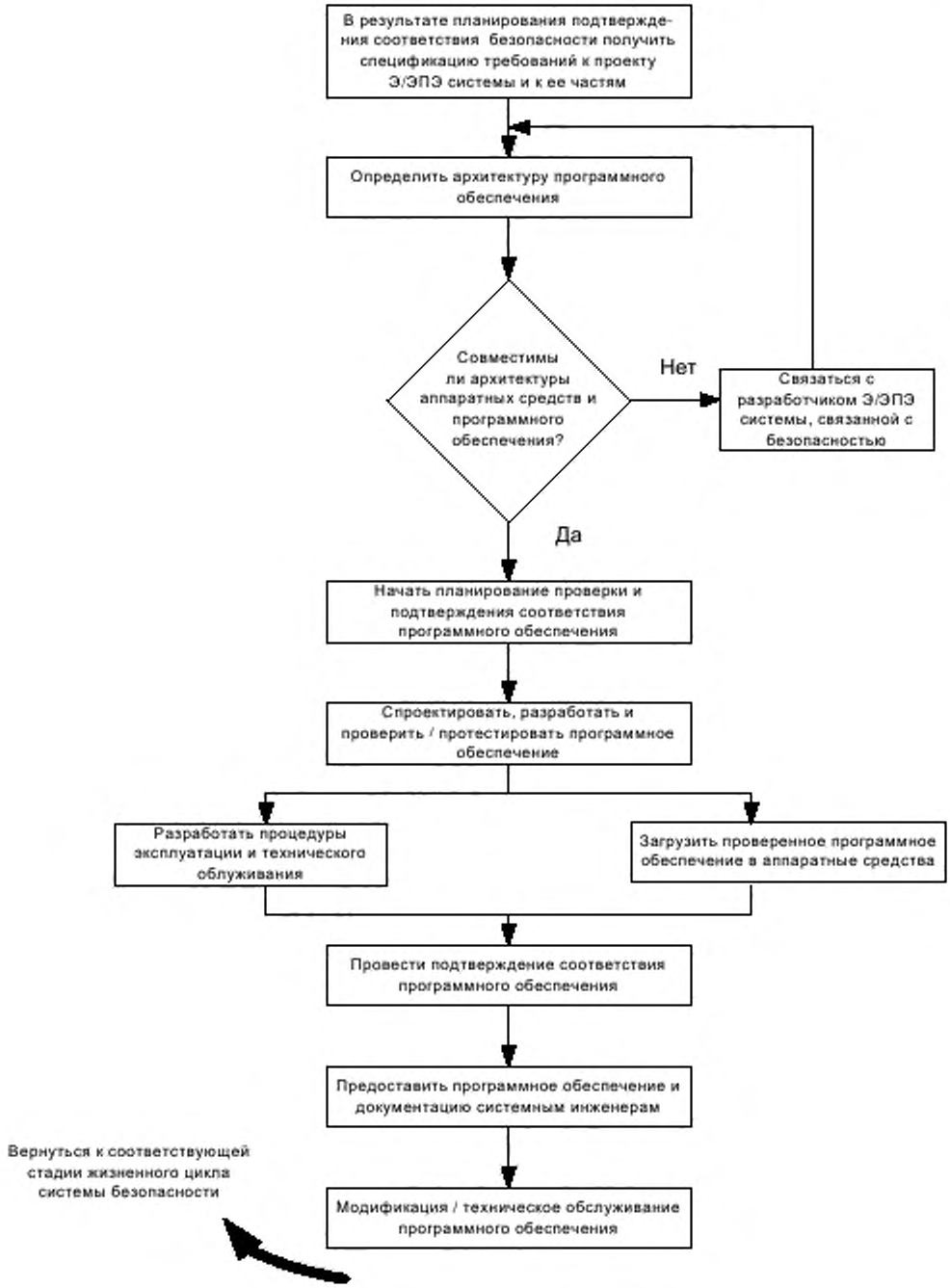


Рисунок А.3 – Функциональные этапы применения МЭК 61508-3

Приложение В
(справочное)

Метод оценки вероятностей отказа аппаратных средств

В.1 Общие положения

Настоящее приложение рассматривает методы расчета вероятностей отказа для Э/Э/ПЭ систем, связанных с безопасностью, указанные в МЭК 61508-1 – МЭК 61508-3. Данная информация носит справочный характер и не должна рассматриваться как единственно возможный метод оценки. Однако в данном приложении описывается относительно простой подход к оценке характеристик Э/Э/ПЭ систем, связанных с безопасностью, и даются руководящие указания по использованию альтернативных методов, взятых из классических способов расчета надежности.

Примечания

1 Архитектуры систем, предоставленные в настоящем стандарте, являются примерами и не должны рассматриваться как исчерпывающие, поскольку существует множество других архитектур, которые также можно использовать.

2 См. [2].

Существует достаточное количество методов более или менее непосредственно применимых для анализа полноты безопасности аппаратного обеспечения Э/Э/ПЭ систем, связанных с безопасностью. Обычно они делятся на группы в соответствии со следующими характеристиками:

- статические (логические) и динамические (состояния/переходы) модели;
- аналитические модели и моделирование на основе метода Монте-Карло;

Логические модели включают в себя все модели, описывающие статические логические связи между элементарными отказами и полным отказом системы. Блок-схемы надежности (см. МЭК 61508-7, С.6.4 и МЭК 61078 [3]) и дерево отказов (см. МЭК 61508-7, В.6.6.5 и В.6.6.9) и МЭК 61025 [4] относятся к логическим моделям.

Модели состояний-переходов включают в себя все модели, описывающие, как система себя ведет (переходит из состояния в состояние) в соответствии с произошедшими событиями (отказами, ремонтами, тестами и т.д.). Сети Маркова (см. МЭК 61508-7, В.6.6.6 и МЭК 61165 [5]), сети Петри (см. МЭК 61508-7, В.2.3.3 и В.6.6.10 и МЭК 62551 [6]) и формальные языки принадлежат к моделям состояний-переходов. Исследуются два марковских подхода: упрощенный подход, основанный на специальной формуле (В.3), и общий подход, позволяющий непосредственный расчет графов Маркова (В.5.2). Если для систем безопасности марковский подход не применим, то

вместо него может быть использован метод Монте-Карло. На современных компьютерах расчет возможен даже для уровня УПБ 4. В подразделах В.5.3 и В.5.4 настоящего приложения даны руководящие указания по применению метода Монте-Карло (см. МЭК 61508-7, В.6.6.8) для моделей поведения, использующих сети Петри и формальные языки моделирования.

Упрощенный подход, который представлен первым, основывается на графическом представлении блок-схемы надежности и специальной формулы Маркова, выведенной из работ Тейлора с учетом относительно консервативных гипотез, описанных в В.3.1.

Все эти методы могут быть использованы для большинства систем, связанных с безопасностью. При определении, какой метод использовать для конкретного применения, очень важно, чтобы пользователь конкретного метода был компетентен в его применении и это, может быть, более важно, чем сам используемый метод. Аналитик отвечает за то, чтобы гипотеза, лежащая в основе любого конкретного метода, была выполнена для рассматриваемого применения либо была внесена какая-либо необходимая корректировка для достижения соответствующего реалистичного консервативного результата. В случае недостаточной надежности данных или превалирующего количества отказов по общей причине может быть достаточным использование простейшей модели/метода. Важна потеря точности или нет, определяется в каждом конкретном случае.

Если для проведения расчетов используется программное обеспечение, то специалист, выполняющий расчет, должен понимать формулы/методы, используемые в программном пакете, чтобы быть уверенным в том, что они применимы в каждом конкретном случае. Специалист также должен проверить программный пакет путем сравнения результатов расчета нескольких тестовых примеров, полученных с помощью программного пакета и ручным способом.

Если отказ системы управления УО инициирует обращение к Э/Э/ПЭ системе, связанной с безопасностью, то вероятность возникновения опасного события зависит также и от вероятности отказа системы управления УО. В этой ситуации необходимо рассмотреть возможность одновременного отказа компонентов системы управления УО и Э/Э/ПЭ системы, связанной с безопасностью, из-за механизмов отказа по общей причине. При неправильном анализе наличие подобных отказов может привести к большим, по сравнению с ожидаемым, значениям остаточного риска.

B.2 Основные вероятностные расчеты

B.2.1 Введение

Блок-схема надежности на рис. B.1 представляет систему (контур) безопасности, состоящую из трех датчиков (A, B, C), одного логического решающего стройства (D), двух исполнительных элементов (E, F) и наличие в ней отказов по общей причине (CCF).

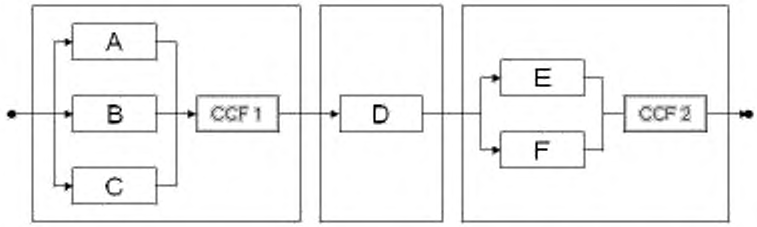


Рисунок B.1 – Блок-схема надежности всей системы безопасности

Эта блок-схема облегчает выявление пяти комбинаций отказов, ведущих к отказу Э/Э/ПЭ системы, связанной с безопасностью. Каждая из них называется минимальным сечением:

- (A, B, C) – тройной отказ
- (E, F) – двойной отказ
- (D), (CCF1), (CCF2) – одиночные отказы

B.2.2 Э/Э/ПЭ система, связанная с безопасностью, с низкой интенсивностью запросов

Когда Э/Э/ПЭ система, связанная с безопасностью, используется в режиме с низкой интенсивностью запросов, стандарт требует, чтобы была дана оценка *PFD*avg (т.е. средняя неготовность, средняя вероятность опасных отказов по запросу). Это просто отношение *MDT*(*T*)/*T*, где *MDT*(*T*) означает время простоя Э/Э/ПЭ системы, связанной с безопаснотью, в период [0, *T*].

Для систем безопасности вероятность отказа обычно очень низка и вероятность одновременного наличия двух минимальных сечений ничтожна. Поэтому суммарное значение средних периодов простоя всех минимальных сечений дает консервативную оценку среднего времени простоя всей системы. Из рис. B.1:

$$MDT \approx MDT^{ABC} + MDT^{D} + MDT^{EF}.$$

Деление на T дает:

$$PFD_{avg} \approx PFD_{avg}^{ABC} + PFD_{avg}^{D} + PFD_{avg}^{EF}.$$

Таким образом, для компонент, соединенных последовательно, вычисления PFD_{avg}

довольно похожи на те, что выполняются с обычной вероятностью, очень малой по сравнению с 1.

Однако для параллельно соединенных компонент, где потеря функциональности возможна только при множественном отказе, таком как (*E*, *F*), очевидно, что MDT^{EF} нельзя вычислить непосредственно из MDT^{E} и MDT^{F}. *MDT* системы (*E*, *F*) должно вычисляться следующим образом:

$$MDT^{EF} = \int_0^T PFD^E(t)PFD^F(t)dt .$$

Таким образом, для параллельно соединенных компонент обычные вероятностные расчеты (на основе сложений и умножений) не подходят для расчета PFD_{avg}, основанного на вычислении указанного интеграла. Свойства PFD_{avg} и обычной вероятности не одни и те же, и объединение этих вероятностей, скорее всего, не приведет к консервативным результатам. В частности, невозможно получить PFD_{avg} Э/Э/ПЭ системы, связанной с безопасностью, только путем объединения обычным способом значений $PFD_{avg,i}$ ее компонент. Так как такой подход иногда предлагается коммерческими логическими программными пакетами, аналитики должны быть внимательны, чтобы избежать таких неконсервативных расчетов, которые нежелательны при обеспечении безопасности.

***Пример** – Для канала с избыточностью (1oo2) с интенсивностью опасных необнаруженных отказов λ_{DU} и интервалом между контрольными испытаниями т расчет по некорректной вероятностной модели может дать $(\lambda_{DU}т)^2/4$, когда в действительности должна быть равна $(\lambda_{DU}т)^2/3$.*

Вычисления могут быть выполнены аналитически или используя метод Монте-Карло. Настоящее приложение описывает, как выполнить эти вычисления, используя общепринятые модели надежности, основанные на логических подходах (блок-схемы надежности или дерево отказов) или моделях состояний-переходов (сети Маркова, сети Петри и т.д.).

B.2.3 Режим работы с непрерывным запросом или режим работы с высокой интенсивностью запросов Э/Э/ПЭ системы, связанной с безопасностью

B.2.3.1 Общая формула *PFH*

Когда Э/Э/ПЭ система, связанная с безопасностью, используется в режиме с непрерывным запросом или с высокой интенсивностью запросов, настоящий стандарт требует вычисления значения *PFH* (т.е. средней частоты опасных отказов). Это среднее значение так называемой безусловной интенсивности отказов (так же называется

частотой отказов) $w(t)$ за интересующий период вычисляется по формуле

$$PFH(T) = \frac{1}{T}\int_0^T w(t)dt \ .$$

Если Э/Э/ПЭ система, связанная с безопасностью, работает в режиме с непрерывным запросом и является основным средством обеспечения безопасности, то отказ всей системы, связанной с безопасностью, ведет непосредственно к потенциально опасной ситуации. Следовательно, при вычислениях можно считать, что отказы, приводящие к отказу функции безопасности всей системы, вся система, связанная с безопасностью, не исправляет. Однако, если отказ всей системы, связанной с безопасностью, не ведет непосредственно к потенциальной опасности при наличии каких-либо других средств безопасности или отказу оборудования, то возможно рассмотреть обнаружение отказа в системе, связанной с безопасностью, и ее ремонт при расчете снижения риска этой системой.

B.2.3.2 Вероятность появления отказа (например в случае единственного средства, работающего в режиме с непрерывным запросом)

Данный случай используется, когда Э/Э/ПЭ система, связанная с безопасностью, работает в режиме с непрерывным запросом и является основным средством обеспечения безопасности. Таким образом, сразу после ее отказа может возникнуть потенциально опасная ситуация. Ни один отказ всей системы не допустим в рассматриваемом периоде.

В этом случае *PFH* может быть рассчитан, используя вероятность появления отказа, деленую на величину рассмотриваемого периода времени:

$$F(T): PFH(T) = \frac{1 - \exp\left[-\int_0^T \lambda(t)dt\right]}{T} = \frac{F(T)}{T} \ .$$

Интенсивность полного отказа системы, $\lambda(t)$, может зависеть от времени или быть константой.

Если она зависит от времени, то: $PFH(T) = \frac{1 - \exp(-\lambda_{as}T)}{T} \approx \lambda_{avg}$.

Если система создана из компонентов, полностью и быстро восстанавливаемых, с постоянными интенсивностями отказов и ремонта (например в случае обнаруживаемых опасных отказов), то $\lambda(t)$ быстро достигает своего постоянного асимптотического значения λ_{as} и, когда $PFH(T)<<1$, имеем следующее значение:

$$PFH(T) = \frac{1 - \exp(-\lambda_{as}T)}{T} \approx \lambda_{avg} = \frac{1}{MTTF} \ .$$

Λ_{as} существует только тогда, когда Э/Э/ПЭ система, связанная с безопасностью, работающая в непрерывном режиме, включает в себя только безопасные и *DD* отказы (например быстро обнаруживаемые и исправимые). Не рассматривается ремонт тех отказов, которые могут привести к полному отказу функции безопасности. Для конфигурации с резервированием, где применимы контрольные проверки, асимптотическая интенсивность отказов не применима и должны быть использованы предыдущие формулы. Выбор конкретного случая выполняет аналитик.

B.2.3.3 Неготовность (например, при наличии нескольких средств обеспечения безопасности)

Когда Э/Э/ПЭ система, связанная с безопасностью, работает в режиме с непрерывным запросом и не является единственным средством обеспечения безопасности, отказы лишь увеличивают частоту запросов к другим средствам обеспечения безопасности, а также, когда она работает в режиме высокой интенсивности запросов и при этом в период ожидания запроса существует возможность выявить (автоматически или вручную) и устранить отказ, который может привести к непосредственному отказу функции безопасности. В этом случае отказы всей системы могут быть исправлены, и *PFH* может быть рассчитан исходя из значений готовности, $A(t)$, и условной интенсивности отказа системы $\Lambda_v(t)$.

Опять же, если система создана из компонентов, которые могут быть полностью и быстро исправлены (например, когда в любой ситуации, ведущей к ухудшению работы, существует большая вероятность быстро вернуть все в нормальное рабочее состояние), $\Lambda_v(t)$, то быстро достигает асимптического значения Λ_{vas}, которая, в дополнение ко всему, является неплохим приближением к действительной асимптоте интенсивности отказа всей системы Λ_{as}, введенной в B.2.3.2.

Это ведет к следующему приближению:

$$PFH = \frac{1}{MUT + MDT} = \frac{1}{MTBF} \approx \frac{1}{MUT} \approx \frac{1}{MTTF},$$

где: *MUT* – среднее значение времени работы (Mean Up Time),

MDT – среднее значение времени простоя (Mean Down Time),

MTBF – среднее время между отказами (Mean Time Between Failure),

MTTF – среднее время до отказа (Mean Time To Failure).

B.2.3.4 Обсуждение интенсивности отказов

Некоторые формулы, приведенные выше, используют интенсивность отказов всей системы $\Lambda(t)$. Ее вычисление не очень простое, поэтому необходимо отметить

следующее.

Для вычисления интенсивности отказов структуры, состоящей из последовательно соединенных компонент, необходимо просто сложить интенсивности отказов каждой из компонент. Для структуры на рисунке В.1 можно написать следующее:

$$\Lambda(t) = \Lambda^{\mathrm{abc}}(t) + \lambda^{\mathrm{CCF1}}(t) + \lambda^{\mathrm{d}}(t) + \Lambda^{\mathrm{ef}}(t) + \lambda^{\mathrm{CCF2}}(t),$$

где $\Lambda(t)$ означает интенсивность полного отказа Э/Э/ПЭ системы, связанной с безопасностью, $\Lambda^{\mathrm{abc}}(t), \lambda^{\mathrm{CCF1}}(t), \lambda^{\mathrm{d}}(t), \Lambda^{\mathrm{ef}}(t), \lambda^{\mathrm{CCF2}}(t)$ являются интенсивностями отказов пяти минимальных сечений.

Для параллельных структур все сложнее, так как в этом случае нет простых соотношений с интенсивностями отказов отдельных компонент. Например, рассмотрим сечение (*E*, *F*):

1) Если *E* и *F* не могут быть мгновенно восстановлены (например в случае *DU* отказа), $\Lambda^{\mathrm{ef}}(t)$ изменяется непрерывно от 0 до λ (интенсивность отказов *E* или *F*). Асимптотическое значение достигается, когда одна из двух компонент вот-вот откажет. Это довольно длительный процесс, т.к. это проявляется, когда *t* становится больше . Данное значение никогда не будет достигнуто, если *E* и *F* периодически проверяют с периодом $\tau \ll {}^{1}/_{\lambda}$.
2) Если *E* и *F* могут быть восстановлены в относительно короткий период времени (например в случае *DD* отказа), $\Lambda^{\mathrm{ef}}(t)$ очень быстро достигает асимптотического значения $\Lambda_{As}^{ef} = \frac{2\lambda^2}{\mu}$, которое может быть использовано как эквивалент постоянной интенсивности отказов. Это значение достигается, когда *t* становится в 2-3 раза больше, чем значения *MTTR* компонентов. Этот особый случай полностью и быстро восстанавливаемых систем описан выше.

Таким образом, в общем случае, оценка интенсивностей отказов всей системы требует более сложных вычислений, чем для более простой последовательной структуры.

В.3 Метод блок-схемы надежности при постоянной интенсивности отказов

В.3.1 Основная гипотеза

Расчеты основываются на следующих предположениях:

- значение результирующей средней вероятности отказа выполнения функции безопасности по запросу для системы меньше 10^{-1} или значение результирующей средней частоты опасного отказа в час для системы меньше 10^{-5} в час;

Примечание – Предположение означает, что такая Э/Э/ПЭ система, связанная с безопасностью, удовлетворяет требованиям стандарта МЭК 61508 и УПБ 1 (см. таблицы 2 и 3 МЭК 61508-1).

- частота отказов компонент постоянна в течение срока службы системы;

- подсистема датчиков (подсистема ввода) состоит из реального датчика(ов) и любых других компонентов и соединительных проводов, вплоть до компонента (компонентов), но его (их) не включая, где сигналы впервые объединяются с помощью процедуры голосования или другой процедуры (например, при конфигурации каналов из двух датчиков, представленной на рисунке В.2);

- логическая подсистема включает в себя компонент (компоненты), в котором(ых) сигналы вначале объединяются, и все другие компоненты, вплоть до тех компонентов включительно, откуда результирующий сигнал(ы) передается(ются) подсистеме исполнительных элементов;

- подсистема исполнительных элементов (подсистема вывода) включает в себя компоненты и соединения, которые обрабатывают исполнительный сигнал(ы), получаемый(ые) от логической подсистемы, а также исполнительный компонент(ы);

- значения частот отказов аппаратных средств, используемых в качестве исходных данных для расчетов и таблиц, задаются для одного канала подсистемы (например при использовании датчиков в виде архитектуры 2oo3 частота отказов задается для одного датчика, а влияние архитектуры 2oo3 рассчитывается дополнительно);

- значения частот отказов и диагностический охват одинаковы для всех каналов в голосующей группе;

- общая частота отказов аппаратных средств канала подсистемы является суммой значений частоты опасных и частоты безопасных отказов для данного канала, которые полагают равными.

Примечание – Это предположение влияет на долю безопасных отказов (см. МЭК 61508-2, приложение С), но доля безопасных отказов не влияет на рассчитанные значения вероятности отказа, приведенные в данном приложении;

- для каждой функции безопасности существуют идеальные средства тестирования и устранения отказов (т.е. все отказы, оставшиеся необнаруженными,

обнаруживаются при тестировании), влияние неидеального тестирования – в соответствии с приложением В, пункт В.3.2.5;

- интервал времени между тестовыми испытаниями должен быть, по крайней мере, на порядок больше, чем *MRT*;

- для каждой подсистемы существует единый интервал времени между тестовыми испытаниями и *MRT*.

- ожидаемый интервал между запросами на выполнение функции безопасности должен быть, по крайней мере, на порядок больше интервала времени между тестовыми испытаниями;

- для всех подсистем, работающих в режиме низкой интенсивности запросов, и для архитектур 1oo2, 1oo2D и 2oo3, работающих в режиме высокой интенсивности запросов и в режиме с непрерывным запросом, доля отказов, заданная охватом диагностикой, обнаруживается и устраняется за *MTTR*, приведенное в требованиях к полноте безопасности аппаратных средств.

Пример – Если предполагаемое MTTR равно 8 ч, то оно включает в себя длительность диагностического тестирования, которое обычно не превышает 1 ч, а оставшаяся часть среднего времени восстановления – это MRT.

Примечание – Для канальных архитектур 1oo2, 1oo2D и 2oo3 предполагается выполнение любого ремонта в оперативном режиме. Если конфигурация Э/Э/ПЭ системы, связанной с безопасностью, при любом обнаруживаемом отказе обеспечивает переход УО в безопасное состояние, то это уменьшает среднюю вероятность отказа при запросе. Степень уменьшения вероятности зависит от охвата диагностикой;

– для канальных архитектур 1oo1 и 2oo2, работающих в режиме высокой интенсивности запросов или в режиме с непрерывным запросом, Э/Э/ПЭ система, связанная с безопасностью, всегда переходит в безопасное состояние после обнаружения опасного отказа; для этого ожидаемый интервал времени между запросами, по крайней мере, должен быть на порядок больше временного интервала диагностического тестирования или сумма временных интервалов диагностического тестирования и временных интервалов перехода в безопасное состояние должна быть меньше, чем время безопасной работы.

Примечание – Время безопасной работы определено в МЭК 61508-4, п. 3.6.20;

– если отказ источника питания приводит к обесточиванию Э/Э/ПЭ системы, связанной с безопасностью, и инициирует переход системы в безопасное состояние, то источник питания не влияет на среднюю вероятность отказа по запросу для Э/Э/ПЭ

системы, связанной с безопасностью; если для перехода в безопасное состояние на систему подается питание или у источника питания существуют режимы отказов, которые могут приводить к небезопасной работе Э/Э/ПЭ системы, связанной с безопасностью, то оценка должна учитывать источник питания;

– если используется терминальный канал, то он ограничивается только той частью рассматриваемой системы, которой обычно являются либо датчик, либо логическая подсистема, либо подсистема исполнительных элементов;

– параметры и их обозначения представлены в таблице В.1.

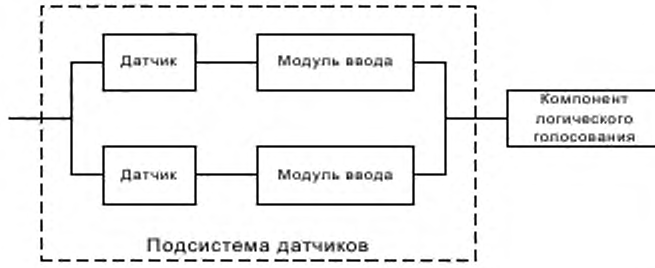


Рисунок В.2 — Пример конфигурации для двух каналов датчиков

Таблица В.1 — Параметры, используемые в настоящем приложении, и диапазоны их значений (применяется к архитектурам 1oo1, 1oo2, 2oo2, 1oo2D и 2oo3)

Обозначение	Параметр, единица измерения	Диапазон параметров в соответствии с таблицами В.2 - В.5 и В.10 - В.13
T_1	Интервал времени между контрольными проверками, ч	Один месяц (730 ч)[1]. Три месяца (2190 ч)[1]. Шесть месяцев (4380 ч). Один год (8760 ч). Два года (17520 ч)[2]. 10 лет (87600 ч)[2].
MTTR	Среднее время восстановления, ч	8 Примечание — *MTTR* = = *MRT* = 8 часов основано на предположении, что время на обнаружение опасного отказа, основанное на автоматическом обнаружении, << *MRT*
MRT	Среднее время ремонта, ч	8 Примечание — *MTTR* = = *MRT* = 8 часов основано на предположении, что время на обнаружение опасного отказа, основанное на автоматическом обнаружении, << *MRT*
DC	Охват диагностикой, дробь (в формулах), % (в остальных случаях)	0 %; 60 %; 90 %; 99 %
β	Доля необнаруженных отказов по общей причине (в таблицах В.2 - В.5 и В.10 - В.13 предполагается $\beta = 2 \times \beta_D$), дробь (в формулах), % (в остальных случаях)	2 %; 10 %; 20 %
β_D	Доля отказов, обнаруженных диагностическими тестами и имеющих общую причину (в таблицах В.2 - В.5 и В.10 - В.13 предполагается $\beta = 2 \times \beta_D$), дробь (в формулах), % (в остальных случаях)	1 %; 5 %; 10 %
λ_{DU}	Интенсивность опасных отказов для канала подсистемы, отказ/ч	$0{,}05 \times 10^{-6}$; $0{,}25 \times 10^{-6}$; $0{,}5 \times 10^{-6}$; $2{,}5 \times 10^{-6}$; 5×10^{-6}; 25×10^{-6}
PFD_G	Средняя вероятность отказа по запросу для группы голосующих каналов (если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной голосующей группы, то PFD_G эквивалентна PFD_S, PFD_L. или PFD_{FE} соответственно)	–
PFD_S	Средняя вероятность отказа по запросу для подсистемы датчиков	–
PFD_L	Средняя вероятность отказа по запросу для логической подсистемы	–
PFD_{FE}	Средняя вероятность отказа по запросу для подсистемы исполнительных элементов	–
PFD_{SYS}	Средняя вероятность отказа по запросу для функции безопасности Э/Э/ПЭ системы, связанной с безопасностью	–

Окончание таблицы В.1

Обозначение	Параметр, единица измерения	Диапазон параметров в соответствии с таблицами В.2 - В.5 и В.10 - В.13
PFH_G	Средняя частота опасных отказов для группы голосующих каналов (если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной голосующей группы, то PFH_G эквивалентна PFH_S, PFH_L, или PFH_{FE} соответственно), отказ/ч	–
PFH_S	Средняя частота опасных отказов для подсистемы датчиков, отказ/ч	–
PFH_L	Средняя частота опасных отказов для логической подсистемы, отказ/ч	–
PFH_{FE}	Средняя частота опасных отказов для подсистемы исполнительных элементов, отказ/ч	–
PFH_{SYS}	Средняя частота опасных отказов для функции безопасности Э/Э/ПЭ системы, связанной с безопасностью, отказ/ч	–
λ	Общая интенсивность отказов для канала подсистемы, отказ/ч	
λ_D	Интенсивность опасных отказов для канала подсистемы, равная $0,5 \times \lambda$ (в предположении 50 % опасных отказов и 50 % безопасных отказов), отказ/ч	–
λ_{DD}	Интенсивность обнаруженных опасных отказов для канала подсистемы (это сумма всех интенсивностей обнаруженных опасных отказов для канала подсистемы), отказ/ч	–
λ_{DU}	Интенсивность необнаруженных опасных отказов для канала подсистемы (это сумма всех интенсивностей необнаруженных опасных отказов для канала подсистемы), отказ/ч	–
λ_{SD}	Интенсивность обнаруженных безопасных отказов для канала подсистемы (это сумма всех интенсивностей обнаруженных безопасных отказов для канала подсистемы), отказ/ч	–
t_{CE}	Эквивалентное среднее время простоя канала для архитектур 1oo1, 1oo2, 2oo2 и 2oo3 (это объединенное время простоя для всех компонентов канала подсистемы), ч	–
t_{GE}	Эквивалентное среднее время простоя голосующей группы для архитектур 1oo1, 1oo2, 2oo2 и 2oo3 (это объединенное время простоя для всех каналов в голосующей группе), ч	–
t_{CE}'	Эквивалентное среднее время простоя канала для архитектуры 1oo2D (это объединенное время простоя для всех компонентов канала подсистемы), ч	–
t_{GE}'	Эквивалентное среднее время простоя голосующей группы для архитектуры 1oo2D (это суммарное время простоя для всех каналов в голосующей группе), ч	–
T_2	Интервал времени между запросами, ч	–
K	Доля успеха при автотестировании схемы в 1oo2D системе	–
PTC	Охват контрольными проверками	–
[1] Только режим высокой интенсивности запросов и режим с непрерывным запроссом. [2] Только режим низкой интенсивности запросов.		

В.3.2 Средняя вероятность отказа по запросу (для режима низкой интенсивности запросов)

В.3.2.1 Процедура расчета

Среднюю вероятность отказа функции безопасности для Э/Э/ПЭ системы,

связанной с безопасностью, определяют вычислением и суммированием средней вероятности отказа в обслуживании для всех подсистем, совокупность которых обеспечивает функцию безопасности. Так как рассматриваемые в настоящем приложении вероятности невелики, то средняя вероятность отказа по запросу для функции безопасности Э/Э/ПЭ системы (см. рисунок В.3), связанной с безопасностью, PFD_{SYS} может быть вычислена по формуле

$$PFD_{SYS} = PFD_S + PFD_L + PFD_{FE} ,$$

где PFD_{SYS} – средняя вероятность отказа по запросу функции безопасности для Э/Э/ПЭ системы, связанной с безопасностью;

PFD_S – средняя вероятность отказа по запросу для подсистемы датчиков;

PFD_L – средняя вероятность отказа по запросу для логической подсистемы;

PFD_{FE} – средняя вероятность отказа по запросу для подсистемы исполнительных элементов.

Рисунок В.3 – Структура подсистем Э/Э/ПЭ системы, связанной с безопасностью

Для определения средней вероятности отказа по запросу для каждой из подсистем необходимо строго придерживаться следующей процедуры для каждой подсистемы:

a) Рисуют структурную схему, изображающую компоненты подсистемы датчиков (подсистемы ввода), компоненты логической подсистемы или компоненты подсистемы исполнительных элементов (подсистемы вывода). Компонентами подсистемы датчиков например могут быть датчики, защитные экраны, входные согласующие цепи; компонентами логической подсистемы – процессоры и сканеры; а компонентами подсистемы исполнительных элементов – выходные согласующие цепи, экраны и исполнительные механизмы. Каждую подсистему представляют как одну либо более голосующих групп 1oo1, 1oo2, 2oo2, 1oo2D или 2oo3.

b) Применяют соответствующие таблицы В.2 – В.5, в которых приведены шестимесячные, годовые, двухлетние и 10-летние интервалы между процедурами тестирования. Данные таблицы предполагают, что среднее время восстановления для

любого отказа после его обнаружения равно 8 ч.

c) Для каждой голосующей группы в подсистеме выбирают из таблиц В.2 – В.5:

- архитектуру (например, 2oo3);
- охват диагностикой для каждого канала (например, 60 %);
- интенсивность опасных отказов (в час) λ_D для каждого канала (например, 2,5E-06);
- β-факторы отказа с общей причиной, β и β_D для взаимосвязи между каналами в рассматриваемой архитектуре (например 2 % и 1 % соответственно).

Примечания

1 Предполагается, что все каналы в голосующей группе имеют одинаковые охват диагностикой и интенсивность отказов (см. В.1).

2 В таблицах В.2 – В.5 (см. также таблицы В.10 – В.13) предполагается, что β-фактор в отсутствие диагностических тестов (также применяемый для необнаруженных опасных отказов при использовании диагностических тестов) β в два раза больше β-фактора для отказов, обнаруживаемых диагностическими тестами, β_D.

d) Получают из таблиц В.2 – В.5 среднюю вероятность отказа для голосующей группы.

e) Если функция безопасности зависит от нескольких голосующих групп датчиков или исполнительных механизмов, то совокупную среднюю вероятность отказа для подсистемы датчиков или подсистемы исполнительных элементов PFD_S или PFD_{FE} задают следующими формулами:

$$PFD_S = \sum_i PFD_{Gi} \quad ,$$

$$PFD_{FE} = \sum_j PFD_{Gj} \quad ,$$

где PFD_{Gi} и PFD_{Gj} – средние вероятности отказа для каждого из голосующей группы датчика или исполнительного элемента, соответственно.

Эти формулы используются во всех уравнениях как для *PFD*, так и для интенсивности отказов системы, которые все являются функцией интенсивности отказов компонентов и среднего времени простоя (*MDT*). Если система состоит из нескольких элементов и требуется определить общую *PFD* комбинации всех элементов или интенсивность отказов системы, то обычно необходимо использовать единое значение *MDT* при вычислениях. Однако каждый элемент может иметь разные механизмы обнаружения отказов с разными *MDT*, так же, как и разные элементы могут иметь разные *MDT* для одних и тех же механизмов обнаружения отказов. В этом случае необходимо вычислить единое значение *MDT*, которое отражает все элементы в

цепочке. Это можно выполнить, рассматривая полные цепочки интенсивности отказов всей системы и пропорционально распределяя индивидуальные значения *MDT* для элементов в соответствии с вкладом их интенсивностей отказов в общую интенсивность отказов.

В качестве примера: если есть два элемента в последовательности, один с интервалом между контрольными проверками T_1, другой с интервалом между контрольными проверками T_2, тогда эквивалентное единое значение для MDT будет равно:

$$\lambda_T = \lambda_1 + \lambda_2,$$

$$MDT_E = \frac{\lambda_1}{\lambda_T}\left(\frac{T_1}{2}\right) + \frac{\lambda_2}{\lambda_T}\left(\frac{T_2}{2}\right).$$

B.3.2.2 Архитектуры для режима низкой интенсивности запросов

Примечания

1 В настоящем пункте справедливые для нескольких архитектур формулы выводят там, где они встречаются впервые.

2 Формулы настоящего пункта справедливы для предположений, перечисленных в B.3.1.

3 Приведенные примеры являются типичными конфигурациями и не являются исчерпывающими.

B.3.2.2.1 Архитектура 1oo1

Данная архитектура предполагает использование одного канала, и любой опасный отказ приводит к нарушению функции безопасности при возникновении запроса на ее выполнение.

Рисунок B.4 – Структурная схема архитектуры 1oo1

На рисунках B.4 и B.5 представлены соответствующие структурные схемы. Интенсивность опасного отказа для канала будет равна:

$$\lambda_D = \lambda_{DU} + \lambda_{DD}\ .$$

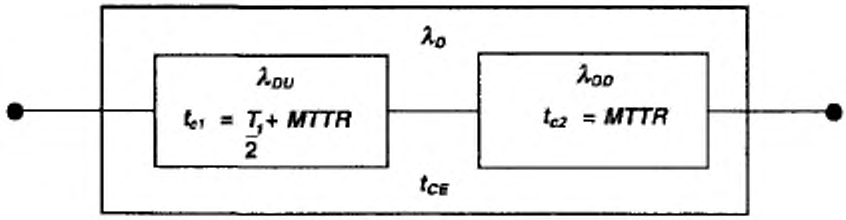


Рисунок В.5 – Структурная схема надежности для архитектуры 1oo1

На рисунке В.5 показано, что канал можно рассматривать как состоящий из двух компонент, одной – с интенсивностью опасных отказов λ_{DU}, обусловленной необнаруженными отказами, а другой – с интенсивностью опасных отказов λ_{DD}, обусловленной обнаруженными отказами. Эквивалентное среднее время простоя канала t_{CE} можно рассчитать, суммируя времена простоя для двух компонент , t_{C1} и t_{C2}, прямо пропорционально вкладу каждой компоненты в вероятность отказа канала:

$$t_{CE} = \frac{\lambda_{DU}}{\lambda_D}\left(\frac{T_1}{2} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR \quad .$$

Для каждой архитектуры интенсивность необнаруженных опасных отказов λ_{DU} и интенсивность необнаруженных опасных отказов λ_{DD} задаются как

$$\lambda_{DU} = \frac{\lambda}{2}(1 - DC) \; ; \; \lambda_{DD} = \frac{\lambda}{2} DC \; .$$

Среднюю вероятность отказа выполнения функции безопасности канала *PFD* в течение времени простоя t_{CE} определяют из выражения

$$PFD = 1 - e^{-\lambda_D t_{CE}} = \lambda_D \, t_{CE} \; , \text{ так как } \lambda_D \, t_{CE} << 1.$$

Следовательно, средняя вероятность отказа по запросу для архитектуры 1oo1 PFD_G равна

$$PFD_G = (\lambda_{DU} + \lambda_{DD}) t_{CE} \; .$$

В.3.2.2.2 Архитектура 1oo2

Данная архитектура представляет собой два канала, соединенных параллельно, так что любой из каналов может выполнить функцию безопасности. Следовательно, для нарушения функции безопасности опасные отказы должны возникнуть в обоих каналах. Предполагается, что любое диагностическое тестирование только сообщает о найденных сбоях и не может изменить ни выходные состояния каналов, ни результат голосования.

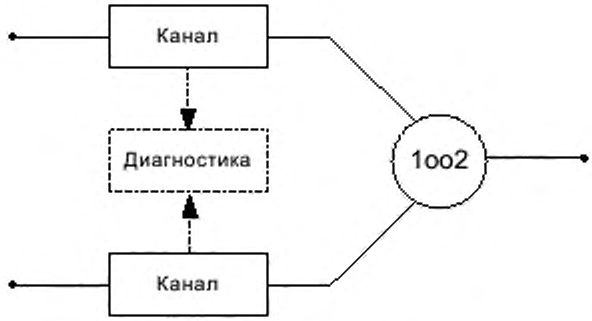


Рисунок В.6 – Структурная схема архитектуры 1oo2

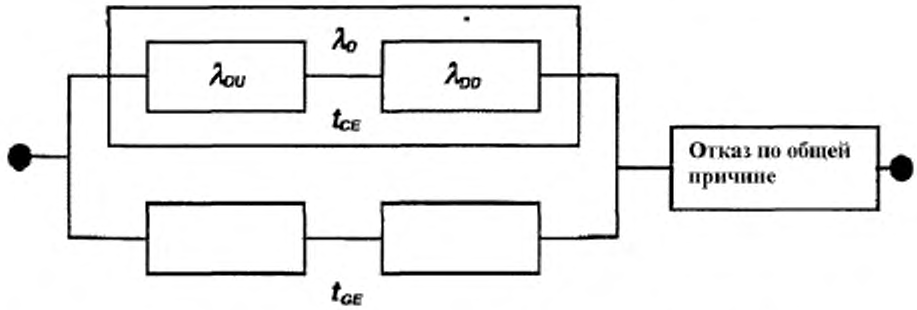


Рисунок В.7 – Структурная схема надежности для архитектуры 1oo2

На рисунках В.6 и В.7 представлены соответствующие структурные схемы. Значение t_{CE} вычисляют в соответствии с В.3.2.2.1, но необходимо вычислить также и эквивалентное время простоя системы t_{GE} по формуле

$$t_{GE} = \frac{\lambda_{DU}}{\lambda_D}\left(\frac{T_1}{3} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR .$$

Для данной архитектуры средняя вероятность отказа по запросу PFD_G равна

$$PFD_G = 2\left((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\right)^2 t_{CE} t_{GE} + \beta_D \lambda_{DD} MTTR + \beta\lambda_{DU}\left(\frac{T_1}{2} + MRT\right).$$

В.3.2.2.3 Архитектура 2oo2

Данная архитектура представляет собой два канала, соединенных параллельно, и для выполнения функции безопасности необходима работа обоих каналов. Предполагается, что любое диагностическое тестирование только сообщает о найденных сбоях и не может изменить ни выходные состояния каналов, ни результат голосования.

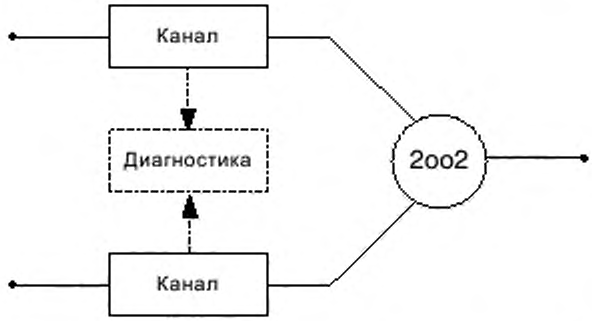


Рисунок В.8 – Структурная схема архитектуры 2oo2

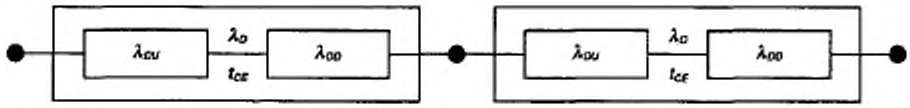


Рисунок В.9 – Структурная схема надежности для архитектуры 2oo2

На рисунках В.8 и В.9 представлены соответствующие структурные схемы. Значение t_{CE} вычисляют в соответствии с В.3.2.2.1, а средняя вероятность отказа по запросу PFD_G для данной архитектуры должна быть равна

$$PFD_G = 2\lambda_D t_{CE} .$$

В.3.2.2.4 Архитектура 1oo2D

Данная архитектура представляет собой два канала, соединенных параллельно. При нормальной работе для выполнения функции безопасности по запросу необходимы оба канала. Кроме того, если диагностическое тестирование обнаруживает отказ в любом канале, то результаты анализа устанавливаются так, чтобы общее выходное состояние совпадало с результатом, выдаваемым другим каналом. Если диагностическое тестирование обнаруживает отказы в обоих каналах или несоответствие между ними, причина которого не может быть идентифицирована, то выходной сигнал переводит систему в безопасное состояние. Для обнаружения несоответствия между каналами каждый канал может определять состояние другого канала независящим от другого канала способом. Сравнение канала/механизма переключения не могут быть 100%-но эффективными, поэтому параметр *K* представляет собой эффективность межканального сравнения/механизма переключения, т.е. выход может оставаться таким же, как и для архитектуры 2oo2, даже если в одном из каналов обнаружен отказ.

Примечание – Параметр *K* необходимо определить с помощью FMEA.

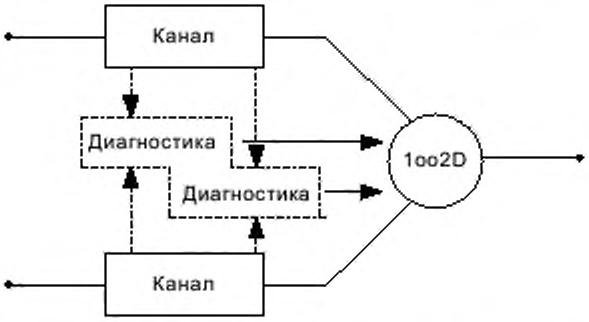


Рисунок В.10 – Структурная схема архитектуры 1oo2D

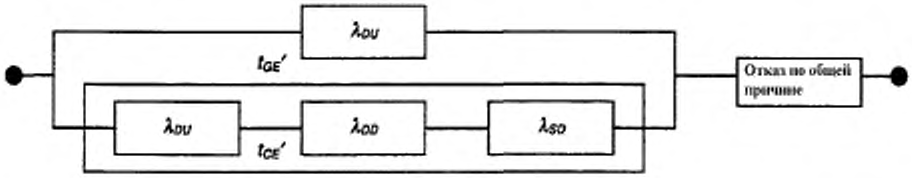


Рисунок В.11 – Структурная схема надежности для архитектуры 1oo2D

Для каждого канала интенсивность обнаруженных безопасных отказов λ_{SD} определяют как

$$\lambda_{SD} = \frac{\lambda}{2} DC .$$

На рисунках В.10 и В.11 представлены соответствующие структурные схемы. Значения эквивалентного среднего времени простоя отличаются от значений, приведенных для других архитектур в В.3.2.2, и поэтому их обозначают как t_{CE}' и t_{GE}'. Эти значения определяют как

$$t_{CE}' = \frac{\lambda_{DU}\left(\frac{T_1}{2} + MRT\right) + (\lambda_{DD} + \lambda_{SD})MTTR}{\lambda_{DU} + (\lambda_{DD} + \lambda_{SD})},$$

$$t_{GE}' = \frac{T_1}{3} + MRT.$$

Средняя вероятность отказа по запросу PFD_G для данной архитектуры равна

$$PFD_G = 2(1-\beta)\lambda_{DU}((1-\beta)\lambda_{DU} + (1-\beta_D)\lambda_{DD} + \lambda_{SD})t_{CE}'t_{GE}' + 2(1-K)\lambda_{DD}t_{CE}' \; + + \beta\lambda_{DU}\left(\frac{T_1}{2} + MRT\right)$$

В.3.2.2.5 Архитектура 2oo3

Данная архитектура состоит из трех каналов, соединенных параллельно с мажорированием выходных сигналов так, что выходное состояние не меняется, если

результат, выдаваемый одним из каналов, отличается от результата, выдаваемого двумя другими каналами.

Предполагается, что любое диагностическое тестирование только фиксирует найденные сбои и не может изменить ни выходные состояния каналов, ни результат голосования.

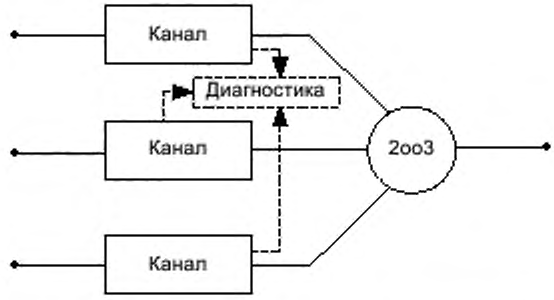


Рисунок В.12 – Структурная схема архитектуры 2oo3

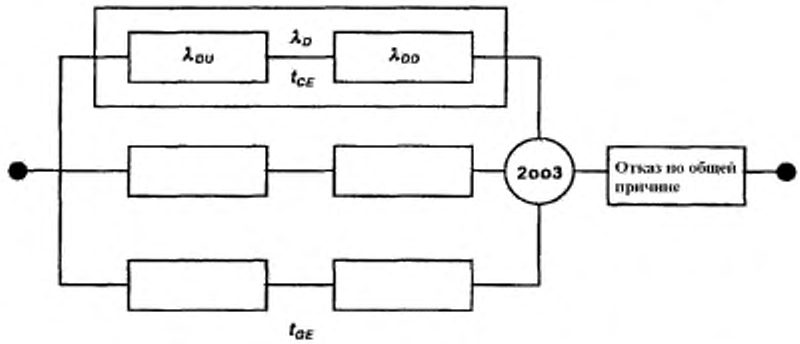


Рисунок В.13 – Структурная схема надежности для архитектуры 2oo3

На рисунках В.12 и В.13 представлены соответствующие структурные схемы. Значение t_{CE} вычисляют по В.3.2.2.1, а значение t_{GE} – по В.3.2.2.2. Средняя вероятность отказа по запросу PFD_G для данной архитектуры равна

$$PFD_G = 6\left((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\right)^2 t_{CE} t_{GE} + \beta_D \lambda_{DD} MTTR + \beta\lambda_{DU}\left(\frac{T_1}{2} + MRT\right).$$

В.3.2.2.6 Архитектура 1oo3

Данная архитектура состоит из трех каналов, соединенных параллельно со схемой голосования для выходных сигналов, так что выходной сигнал соответствует схеме голосования 1oo3.

Предполагается, что любая диагностическая проверка только сообщает о найденных отказах и не меняет никаких выходных состояний или выхода схемы голосования.

Значение t_{CE} вычисляют по В.3.2.2.1, а значение t_{GE} – по В.3.2.2.2. Средняя вероятность отказа по запросу PFD_G для данной архитектуры равна

$$PFD_G = 6\left((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\right)^3 t_{CE} t_{GE} t_{G2E} + \beta_D \lambda_{DD} MTTR + \beta\lambda_{DU}\left(\frac{T_1}{2}+MRT\right),$$

где $t_{G2E} = \frac{\lambda_{DU}}{\lambda_D}\left(\frac{T_1}{4}+MRT\right)+\frac{\lambda_{DD}}{\lambda_D}MTTR$.

B.3.2.3 Подробные таблицы для режима низкой интенсивности запросов

Таблица B.2 — Средняя вероятность отказа по запросу для шестимесячного интервала между контрольными проверками при среднем времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	1.1 E-04			5.5E-04			1.1E-03		
	60 %	4.4E-05			2.2E-04			4.4E-04		
	90 %	1.1 E-05			5.7E-05			1.1E-04		
	99 %	1.5E-06			7.5E-06			1.5E-05		
1oo2	0 %	2.2E-06	1.1E-05	2.2E-05	1.1E-05	5.5E-05	1.1E-04	2.4E-05	1.1E-04	2.2E-04
	60 %	8.8E-07	4.4E-06	8.8E-06	4.5E-06	2.2E-05	4.4E-05	9.1E-06	4.4E-05	8.8E-05
	90 %	2.2E-07	1.1E-06	2.2E-06	1.1E-06	5.6E-06	1.1E-05	2.3E-06	1.1E-05	2.2E-05
	99 %	2.6E-08	1.3E-07	2.6E-07	1.3E-07	6.5E-07	1.3E-06	2.6E-07	1.3E-06	2.6E-06
2oo2 (см. примечание 2)	0 %	2.2E-04			1.1E-03			2.2E-03		
	60 %	8.8E-05			4.4E-04			8.8E-04		
	90 %	2.3E-05			1.1 E-04			2.3E-04		
	99 %	3.0E-06			1.5E-05			3.0E-05		
1oo2D (см. примечание 3)	0 %	2.2E-06	1.1E-05	2.2E-05	1.1E-05	5.5E-05	1.1E-04	2.4E-05	1.1E-04	2.2E-04
	60 %	1.4E-06	4.9E-06	9.3E-06	7.1E-06	2.5E-05	4.7E-05	1.4E-05	5.0E-05	9.3E-05
	90 %	4.3E-07	1.3E-06	2.4E-06	2.2E-06	6.6E-06	1.2E-05	4.3E-06	1.3E-05	2.4E-05
	99 %	6.0E-08	1.5E-07	2.6E-07	3.0E-07	7.4E-07	1.3E-06	6.0E-07	1.5E-06	2.6E-06
2oo3	0 %	2.2E-06	1.1E-05	2.2E-05	1.2E-05	5.6E-05	1.1E-04	2.7E-05	1.1E-04	2.2E-04
	60 %	8.9E-07	4.4E-06	8.8E-06	4.6E-06	2.2E-05	4.4E-05	9.6E-06	4.5E-05	8.9E-05
	90 %	2.2E-07	1.1E-06	2.2E-06	1.1E-06	5.6E-06	1.1E-05	2.3E-06	1.1E-05	2.2E-05
	99 %	2.6E-08	1.3E-07	2.6E-07	1.3E-07	6.5E-07	1.3E-06	2.6E-07	1.3E-06	2.6E-06
1oo3	0 %	2.2E-06	1.1E-05	2.2E-05	1.1E-05	5.5E-05	1.1E-04	2.2E-05	1.1E-04	2.2E-04
	60 %	8.8E-07	4.4E-06	8.8E-06	4.4E-06	2.2E-05	4.4E-05	8.8E-06	4.4E-05	8.8E-05
	90 %	2.2E-07	1.1E-06	2.2E-06	1.1E-06	5.6E-06	1.1E-05	2.2E-06	1.1E-05	2.2E-05
	99 %	2.6E-08	1.3E-07	2.6E-07	1.3E-07	6.5E-07	1.3E-06	2.6E-07	1.3E-06	2.6E-06

Окончание таблицы В.2

Архитектура	DC	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	5.5E-03			1.1E-02			5.5E-02		
	60 %	2.2E-03			4.4E-03			2.2E-02		
	90 %	5.7E-04			1.1E-03			5.7E-03		
	99 %	7.5E-05			1.5E-04			7.5E-04		
1oo2	0 %	1.5E-04	5.8E-04	1.1E-03	3.7E-04	1.2E-03	2.3E-03	5.0E-03	8.8E-03	1.4E-02
	60 %	5.0E-05	2.3E-04	4.5E-04	1.1E-04	4.6E-04	9.0E-04	1.1E-03	2.8E-03	4.9E-03
	90 %	1.2E-05	5.6E-05	1.1E-04	2.4E-05	1.1E-04	2.2E-04	1.5E-04	6.0E-04	1.2E-03
	99 %	1.3E-06	6.5E-06	1.3E-05	2.6E-06	1.3E-05	2.6E-05	1.4E-05	6.6E-05	1.3E-04
2oo2 (см. примечание 2)	0 %	1.1E-02			2.2E-02			>1E-01		
	60 %	4.4E-03			8.8E-03			4.4E-02		
	90 %	1.1E-03			2.3E-03			1.1E-02		
	99 %	1.5E-04			3.0E-04			1.5E-03		
1oo2D (см. примечание 3)	0 %	1.5E-04	5.8E-04	1.1E-03	3.8E-04	1.2E-03	2.3E-03	5.0E-03	9.0E-03	1.4E-02
	60 %	7,7E-05	2,5E-04	4,7E-04	1,7E-04	5,2E-04	9,5E-04	1,3E-03	3,0E-03	5,1E-03
	90 %	2,2E-05	6,6E-05	1,2E-04	4,5E-05	1,3E-04	2,4E-04	2,6E-04	6,9E-04	1,2E-03
	99 %	3,0E-06	7,4E-06	1,3E-05	6,0E-06	1,5E-05	2,6E-05	3,0E-05	7,4E-05	1,3E-04
2oo3	0 %	2.3E-04	6.5E-04	1.2E-03	6.8E-04	1.5E-03	2.5E-03	1.3E-02	1.5E-02	1.9E-02
	60 %	6.3E-05	2.4E-04	4.6E-04	1.6E-04	5.1E-04	9.4E-04	2.3E-03	3.9E-03	5.9E-03
	90 %	1.2E-05	5.7E-05	1.1E-04	2.7E-05	1.2E-04	2.3E-04	2.4E-04	6.8E-04	1.2E-03
	99 %	1.3E-06	6.5E-06	1.3E-05	2.7E-06	1.3E-05	2.6E-05	1.5E-05	6.7E-05	1.3E-04
1oo3	0 %	1.1E-04	5.5E-04	1.1E-03	2.2E-04	1.1E-03	2.2E-03	1.4E-02	5.7E-03	1.1E-02
	60 %	4.4E-05	2.2E-04	4.4E-04	8.8E-05	4.4E-04	8.8E-04	4.6E-04	2.2E-03	4.4E-03
	90 %	1.1E-05	5.6E-05	1.1E-04	2.2E-05	1.1E-04	2.2E-04	1.1E-04	5.6E-04	1.1E-03
	99 %	1.3E-06	6.5E-06	1.3E-05	2.6E-06	1.3E-05	2.6E-05	1.3E-05	6.5E-05	1.3E-04

Примечания

1 В настоящей таблице приведены примеры значений PFD_G, рассчитанные по формулам в соответствии с В.3.2 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной группы голосующих каналов, то PFD_G эквивалентна PFD_S, PFD_L, или PFD_{FE} соответственно (см. В.3.2.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

Таблица В.3 – Средняя вероятность отказа по запросу для одногодичного интервала между контрольными испытаниями и среднего времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	2.2E-04			1.1 E-03			2.2E-03		
	60 %	8.8E-05			4.4E-04			8.8E-04		
	90 %	2.2E-05			1.1 E-04			2.2E-04		
	99 %	2.6E-06			1.3E-05			2.6E-05		
1oo2	0 %	4.4E-06	2.2E-05	4.4E-05	2.3E-05	1.1E-04	2.2E-04	5.0E-05	2.2E-04	4.4E-04
	60 %	1.8E-06	8.8E-06	1.8E-05	9.0E-06	4.4E-05	8.8E-05	1.9E-05	8.9E-05	1.8E-04
	90 %	4.4E-07	2.2E-06	4.4E-06	2.2E-06	1.1 E-05	2.2E-05	4.5E-06	2.2E-05	4.4E-05
	99 %	4.8E-08	2.4E-07	4.8E-07	2.4E-07	1.2E-06	2.4E-06	4.8E-07	2.4E-06	4.8E-06
2oo2 (см. примечание 2)	0 %	4.4E-04			2.2E-03			4.4E-03		
	60 %	1.8E-04			8.8E-04			1.8E-03		
	90 %	4.5E-05			2.2E-04			4.5E-04		
	99 %	5.2E-06			2.6E-05			5.2E-05		
1oo2D (см. примечание 3)	0 %	4.5E-06	2.2E-05	4.4E-05	2.4E-05	1.1E-04	2.2E-04	5.0E-05	2.2E-04	4.4E-04
	60 %	2,8E-06	9,8E-06	1,9E-05	1,4E-05	4,9E-05	9,3E-05	2,9E-05	9,9E-05	1,9E-04
	90 %	8,5E-07	2,6E-06	4,8E-06	4,3E-06	1,3E-05	2,4E-05	8,5E-06	2,6E-05	4,8E-05
	99 %	1,0E-07	2,8E-07	5,0E-07	5,2E-07	1,4E-06	2,5E-06	1,0E-06	2,8E-06	5,0E-06
2oo3	0 %	4.6E-06	2.2E-05	4.4E-05	2.7E-05	1.1E-04	2.2E-04	6.2E-05	2.4E-04	4.5E-04
	60 %	1.8E-06	8.8E-06	1.8E-05	9.5E-06	4.5E-05	8.8E-05	2.1E-05	9.1E-05	1.8E-04
	90 %	4.4E-07	2.2E-06	4.4E-06	2.3E-06	1.1 E-05	2.2E-05	4.6E-06	2.2E-05	4.4E-05
	99 %	4.8E-08	2.4E-07	4.8E-07	2.4E-07	1.2E-06	2.4E-06	4.8E-07	2.4E-06	4.8E-06
1oo3	0 %	4.4E-06	2.2E-05	4.4E-05	2.2E-05	1.1E-04	2.2E-04	4.4E-05	2.2E-04	4.4E-04
	60 %	1.8E-06	8.8E-06	1.8E-05	8.8E-06	4.4E-05	8.8E-05	1.8E-05	8.8E-05	1.8E-04
	90 %	4.4E-07	2.2E-06	4.4E-06	2.2E-06	1.1 E-05	2.2E-05	4.4E-06	2.2E-05	4.4E-05
	99 %	4.8E-08	2.4E-07	4.8E-07	2.4E-07	1.2E-06	2.4E-06	4.8E-07	2.4E-06	4.8E-06

Окончание таблицы В.3

Архитектура	*DC*	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	1.1 E-02			2.2E-02			>1E-01		
	60 %	4.4E-03			8.8E-03			4.4E-02		
	90 %	1.1 E-03			2.2E-03			1.1 E-02		
	99 %	1.3E-04			2.6E-04			1.3E-03		
1oo2	0 %	3.7E-04	1.2E-03	2.3E-03	1.1E-03	2.7E-03	4.8E-03	1.8E-02	2.4E-02	3.2E-02
	60 %	1.1E-04	4.6E-04	9.0E-04	2.8E-04	9.7E-04	1.8E-03	3.4E-03	6.6E-03	1.1E-02
	90 %	2.4E-05	1.1E-04	2.2E-04	5.1 E-05	2.3E-04	4.5E-04	3.8E-04	1.3E-03	2.3E-03
	99 %	2.4E-06	1.2E-05	2.4E-05	4.9E-06	2.4E-05	4.8E-05	2.6E-05	1.2E-04	2.4E-04
2oo2 (см. примечание 2)	0 %	2.2E-02			4.4E-02			>1E-01		
	60 %	8.8E-03			1.8E-02			8.8E-02		
	90 %	2.2E-03			4.5E-03			2.2E-02		
	99 %	2.6E-04			5.2E-04			2.6E-03		
1oo2D (см. примечание 3)	0 %	3.8E-04	1.2E-03	2.3E-03	1.1E-03	2.7E-03	4.9E-03	1.8E-02	2.5E-02	3.4E-02
	60 %	1,7E-04	5,1E-04	9,5E-04	3,8E-04	1,1E-03	1,9E-03	3,9E-03	7,1E-03	1,1E-02
	90 %	4,4E-05	1,3E-04	2,4E-04	9,1E-05	2,7E-04	4,8E-04	5,8E-04	1,4E-03	2,5E-03
	99 %	5,2E-06	1,4E-05	2,5E-05	1,0E-05	2,8E-05	5,0E-05	5,4E-05	1,4E-04	2,5E-04
2oo3	0 %	6.8E-04	1.5E-03	2.5E-03	2.3E-03	3.8E-03	5.6E-03	4.8E-02	5.0E-02	5.3E-02
	60 %	1.6E-04	5.1E-04	9.4E-04	4.8E-04	1.1 E-03	2.0E-03	8.4E-03	1.1 E-02	1.5E-02
	90 %	2.7E-05	1.2E-04	2.3E-04	6.4E-05	2.4E-04	4.6E-04	7.1 E-04	1.6E-03	2.6E-03
	99 %	2.5E-06	1.2E-05	2.4E-05	5.1E-06	2.4E-05	4.8E-05	3.1E-05	1.3E-04	2.5E-04
1oo3	0 %	2.2E-04	1.1E-03	2.2E-03	4.6E-04	2.2E-03	4.4E-03	4.7E-02	1.3E-02	2.3E-02
	60 %	8.8E-05	4.4E-04	8.8E-04	1.8E-04	8.8E-04	1.8E-03	1.0E-03	4.5E-03	8.9E-03
	90 %	2.2E-05	1.1E-04	2.2E-04	4.4E-05	2.2E-04	4.4E-04	2.2E-04	1.1E-03	2.2E-03
	99 %	2.4E-06	1.2E-05	2.4E-05	4.8E-06	2.4E-05	4.8E-05	2.4E-05	1.2E-04	2.4E-04

Примечания

1 В настоящей таблице приведены примеры значений PFD_G, рассчитанные по формулам в соответствии с В.3.2 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной группы голосующих каналов, то PFD_G эквивалентна PFD_S, PFD_L. или PFD_{FE} соответственно (см. В.3.2.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

Таблица В.4 — Средняя вероятность отказа по запросу для двухлетнего интервала между контрольными испытаниями и среднего времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	4.4E-04			2.2E-03			4.4E-03		
	60 %	1.8E-04			8.8E-04			1.8E-03		
	90 %	4.4E-05			2.2E-04			4.4E-04		
	99 %	4.8E-06			2.4E-05			4.8E-05		
1oo2	0 %	9.0E-06	4.4E-05	8.8E-05	5.0E-05	2.2E-04	4.4E-04	1.1 E-04	4.6E-04	8.9E-04
	60 %	3.5E-06	1.8E-05	3.5E-05	1.9E-05	8.9E-05	1.8E-04	3.9E-05	1.8E-04	3.5E-04
	90 %	8.8E-07	4.4E-06	8.8E-06	4.5E-06	2.2E-05	4.4E-05	9.1 E-06	4.4E-05	8.8E-05
	99 %	9.2E-08	4.6E-07	9.2E-07	4.6E-07	2.3E-06	4.6E-06	9.2E-07	4.6E-06	9.2E-06
2oo2 (см. примечание 2)	0 %	8.8E-04			4.4E-03			8.8E-03		
	60 %	3.5E-04			1.8E-03			3.5E-03		
	90 %	8.8E-05			4.4E-04			8.8E-04		
	99 %	9.6E-06			4.8E-05			9.6E-05		
1oo2D (см. примечание 3)	0 %	9.0E-06	4.4E-05	8.8E-05	5.0E-05	2.2E-04	4.4E-04	1.1 E-04	4.6E-04	9.0E-04
	60 %	5,7E-06	2,0E-05	3,7E-05	2,9E-05	9,9E-05	1,9E-04	6,0E-05	2,0E-04	3,7E-04
	90 %	1,7E-06	5,2E-06	9,6E-06	8,5E-06	2,6E-05	4,8E-05	1,7E-05	5,2E-05	9,6E-05
	99 %	1,9E-07	5,4E-07	9,8E-07	9,5E-07	2,7E-06	4,9E-06	1,9E-06	5,4E-06	9,8E-06
2oo3	0 %	9.5E-06	4.4E-05	8.8E-05	6.2E-05	2.3E-04	4.5E-04	1.6E-04	5.0E-04	9.3E-04
	60 %	3.6E-06	1.8E-05	3.5E-05	2.1 E-05	9.0E-05	1.8E-04	4.7E-05	1.9E-04	3.6E-04
	90 %	8.9E-07	4.4E-06	8.8E-06	4.6E-06	2.2E-05	4.4E-05	9.6E-06	4.5E-05	8.9E-05
	99 %	9.2E-08	4.6E-07	9.2E-07	4.6E-07	2.3E-06	4.6E-06	9.3E-07	4.6E-06	9.2E-06
1oo3	0 %	8.8E-06	4.4E-05	8.8E-05	4.4E-05	2.2E-04	4.4E-04	8.8E-05	4.4E-04	8.8E-04
	60 %	3.5E-06	1.8E-05	3.5E-05	1.8E-05	8.8E-05	1.8E-04	3.5E-05	1.8E-04	3.5E-04
	90 %	8.8E-07	4.4E-06	8.8E-06	4.4E-06	2.2E-05	4.4E-05	8.8E-06	4.4E-05	8.8E-05
	99 %	9.2E-08	4.6E-07	9.2E-07	4.6E-07	2.3E-06	4.6E-06	9.2E-07	4.6E-06	9.2E-06

Окончание таблицы В.4

Архитектура	*DC*	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	2.2E-02			4.4E-02			>1E-01		
	60 %	8.8E-03			1.8E-02			8.8E-02		
	90 %	2.2E-03			4.4E-03			2.2E-02		
	99 %	2.4E-04			4.8E-04			2.4E-03		
1oo2	0 %	1.1 E-03	2.7E-03	4.8E-03	3.3E-03	6.5E-03	1.0E-02	6.6E-02	7.4E-02	8.5E-02
	60 %	2.8E-04	9.7E-04	1.8E-03	7.5E-04	2.1 E-03	3.8E-03	1.2E-02	1.8E-02	2.5E-02
	90 %	5.0E-05	2.3E-04	4.5E-04	1.1 E-04	4.6E-04	9.0E-04	1.1E-03	2.8E-03	4.9E-03
	99 %	4.7E-06	2.3E-05	4.6E-05	9.5E-06	4.6E-05	9.2E-05	5.4E-05	2.4E-04	4.6E-04
2oo2 (см. примечание 2)	0 %	4.4E-02			8.8E-02			>1E-01		
	60 %	1.8E-02			3.5E-02			>1E-01		
	90 %	4.4E-03			8.8E-03			4.4E-02		
	99 %	4.8E-04			9.6E-04			4.8E-03		
1oo2D (см. примечание 3)	0 %	1.1 E-03	2.7E-03	4.8E-03	3.4E-03	6.6E-03	1.1E-02	6.7E-02	7.7E-02	9.0E-02
	60 %	3,8E-04	1,1E-03	1,9E-03	9,6E-04	2,3E-03	4,0E-03	1,3E-02	1,9E-02	2,6E-02
	90 %	9,0E-05	2,6E-04	4,8E-04	1,9E-04	5,4E-04	9,8E-04	1,5E-03	3,2E-03	5,3E-03
	99 %	9,6E-06	2,7E-05	4,9E-05	1,9E-05	5,4E-05	9,8E-05	1,0E-04	2,8E-04	5,0E-04
2oo3	0 %	2.3E-03	3.7E-03	5.6E-03	8.3E-03	1.1 E-02	1.4E-02	1.9E-01	1.8E-01	1.7E-01
	60 %	4.8E-04	1.1E-03	2.0E-03	1.6E-03	2.8E-03	4.4E-03	3.2E-02	3.5E-02	4.0E-02
	90 %	6.3E-05	2.4E-04	4.6E-04	1.6E-04	5.1 E-04	9.4E-04	2.4E-03	4.0E-03	6.0E-03
	99 %	4.8E-06	2.3E-05	4.6E-05	1.0E-05	4.7E-05	9.2E-05	6.9E-05	2.5E-04	4.8E-04
1oo3	0 %	4.6E-04	2.2E-03	4.4E-03	1.0E-03	4.5E-03	8.9E-03	2.4E-02	3.7E-02	5.5E-02
	60 %	1.8E-04	8.8E-04	1.8E-03	3.6E-04	1.8E-03	3.5E-03	3.1E-03	9.9E-03	1.8E-02
	90 %	4.4E-05	2.2E-04	4.4E-04	8.8E-05	4.4 E-04	8.8E-04	4.6E-04	2.2E-03	4.4E-03
	99 %	4.6E-06	2.3E-05	4.6E-05	9.2E-06	4.6E-05	9.2E-05	4.6E-05	2.3E-04	4.6E-04

Примечания

1 В настоящей таблице приведены примеры значений PFD_G, рассчитанные по формулам в соответствии с В.3.2 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной группы голосующих каналов, то PFD_G эквивалентна PFD_S, PFD_L. или PFD_{FE} соответственно (см. В.3.2.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

Таблица В.5 – Средняя вероятность отказа по запросу для десятилетнего интервала между контрольными испытаниями и среднего времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %
1oo1 (см. примечание 2)	0 %	2.2E-03			1.1 E-02			2.2E-02		
	60 %	8.8E-04			4.4E-03			8.8E-03		
	90 %	2.2E-04			1.1E-03			2.2E-03		
	99 %	2.2E-05			1.1 E-04			2.2E-04		
1oo2	0 %	5.0E-05	2.2E-04	4.4E-04	3.7E-04	1.2E-03	2.3E-03	1.1E-03	2.7E-03	4.8E-03
	60 %	1.9E-05	8.9E-05	1.8E-04	1.1E-04	4.6E-04	9.0E-04	2.7E-04	9.6E-04	1.8E-03
	90 %	4.4E-06	2.2E-05	4.4E-05	2.3E-05	1.1 E-04	2.2E-04	5.0E-05	2.2E-04	4.4E-04
	99 %	4.4E-07	2.2E-06	4.4E-06	2.2E-06	1.1 E-05	2.2E-05	4.5E-06	2.2E-05	4.4E-05
2oo2 (см. примечание 2)	0 %	4.4E-03			2.2E-02			4.4E-02		
	60 %	1.8E-03			8.8E-03			1.8E-02		
	90 %	4.4E-04			2.2E-03			4.4E-03		
	99 %	4.5E-05			2.2E-04			4.5E-04		
1oo2D (см. примечание 3)	0 %	5.0E-05	2.2E-04	4.4E-04	3.7E-04	1.2E-03	2.3E-03	1.1 E-03	2.7E-03	4.8E-03
	60 %	2,9E-05	9,9E-05	1,9E-04	1,7E-04	5,1E-04	9,5E-04	3.8E-04	1,1E-03	1,9E-03
	90 %	8,4E-06	2,6E-05	4,8E-05	4,3E-05	1,3E-04	2,4E-04	9,0E-05	2,6E-04	4,8E-04
	99 %	8,9E-07	2,6E-06	4,8E-06	4,5E-06	1,3E-05	2,4E-05	8,9E-06	2,6E-05	4,8E-05
2oo3	0 %	6.2E-05	2.3E-04	4.5E-04	6.8E-04	1.5E-03	2.5E-03	2.3E-03	3.7E-03	5.6E-03
	60 %	2.1 E-05	9.0E-05	1.8E-04	1.6E-04	5.0E-04	9.3E-04	4.7E-04	1.1 E-03	2.0E-03
	90 %	4.6E-06	2.2E-05	4.4E-05	2.7E-05	1.1 E-04	2.2E-04	6.3E-05	2.4E-04	4.5E-04
	99 %	4.4E-07	2.2E-06	4.4E-06	2.3E-06	1.1 E-05	2.2E-05	4.6E-06	2.2E-05	4.4E-05
1oo3	0 %	4.4E-05	2.2E-04	4.4E-04	2.2E-04	1.1E-03	2.2E-03	4.6E-04	2.2E-03	4.4E-03
	60 %	1.8E-05	8.8E-05	1.8E-04	8.8E-05	4.4E-04	8.8E-04	1.8E-04	8.8E-04	1.8E-03
	90 %	4.4E-06	2.2E-05	4.4E-05	2.2E-05	1.1 E-04	2.2E-04	4.4E-05	2.2E-04	4.4E-04
	99 %	4.4E-07	2.2E-06	4.4E-06	2.2E-06	1.1 E-05	2.2E-05	4.4E-06	2.2E-05	4.4E-05

Окончание таблицы В.5

Архитектура	*DC*	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	>1E-01			>1E-01			>1E-01		
	60 %	4.4E-02			8.8E-02			>1E-01		
	90 %	1.1E-02			2.2E-02			>1E-01		
	99 %	1.1 E-03			2.2E-03			1.1 E-02		
1oo2	0 %	1.8E-02	2.4E-02	3.2E-02	6.6E-02	7.4E-02	8.5E-02	>1E-01	>1E-01	>1E-01
	60 %	3.4E-03	6.6E-03	1.1E-02	1.2E-02	1.8E-02	2.5E-02	>1E-01	>1E-01	>1E-01
	90 %	3.8E-04	1.2E-03	2.3E-03	1.1 E-03	2.8E-03	4.9E-03	1.8E-02	2.5E-02	3.5E-02
	99 %	2.4E-05	1.1 E-04	2.2E-04	5.1 E-05	2.3E-04	4.5E-04	3.8E-04	1.3E-03	2.3E-03
2oo2 (см. примечание 2)	0 %	>1E-01			>1E-01			>1E-01		
	60 %	8.8E-02			>1E-01			>1E-01		
	90 %	2.2E-02			4.4E-02			>1E-01		
	99 %	2.2E-03			4.5E-03			2.2E-02		
1oo2D (см. примечание 3)	0 %	1,8E-02	2,5E-02	3,3E-02	6,6E-02	7,7E-02	9,0E-02	1,6E+00	1,5E+00	1,4E+00
	60 %	3,9E-03	7,1E-03	1,1E-02	1,3E-02	1,9E-02	2,6E-02	2,6E-01	2,7E-01	2,8E-01
	90 %	5,7E-04	1,4E-03	2,5E-03	1,5E-03	3,1E-03	5,2E-03	2,0E-02	2,7E-02	3,5E-02
	99 %	4,6E-05	1,3E-04	2,4E-04	9,5E-05	2,7E-04	4,9E-04	6,0E-04	1,5E-03	2,5E-03
2oo3	0 %	4.8E-02	5.0E-02	5.3E-02	1.9E-01	1.8E-01	1.7E-01	4.6E+00	4.0E+00	3.3E+00
	60 %	8.3E-03	1.1 E-02	1.4E-02	3.2E-02	3.5E-02	4.0E-02	7.6E-01	7.1E-01	6.6E-01
	90 %	6.9E-04	1.5E-03	2.6E-03	2.3E-03	3.9E-03	5.9E-03	4.9E-02	5.4E-02	6.0E-02
	99 %	2.7E-05	1.2E-04	2.3E-04	6.4E-05	2.4E-04	4.6E-04	7.1 E-04	1.6E-03	2.6E-03
1oo3	0 %	4.7E-02	1.3E-02	2.3E-02	2.4E-02	3.7E-02	5.5E-02	2.5E+00	2.0E+00	1.6E+00
	60 %	1.0E-03	4.5E-03	8.9E-03	3.0E-03	9.8E-03	1.8E-02	1.7E-01	1.8E-01	1.9E-01
	90 %	2.2E-04	1.1E-03	2.2E-03	4.6E-04	2.2E-03	4.4E-03	4.8E-03	1.3E-02	2.4E-02
	99 %	2.2E-05	1.1E-04	2.2E-04	4.4E-05	2.2E-04	4.4E-04	2.2E-04	1.1E-03	2.2E-03

Примечания

1 В настоящей таблице приведены примеры значений PFD_G, рассчитанные по формулам в соответствии с В.3.2 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной группы голосующих каналов, то PFD_G эквивалентна PFD_S, PFD_L, или PFD_{FE} соответственно (см. В.3.2.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

В.3.2.4. Пример режима низкой интенсивности запросов

Рассмотрим функцию безопасности, для реализации которой нужна система УПБ 2. Пусть построенный на основе предыдущего опыта первоначальный вариант архитектуры всей системы включает одну группу из трех аналоговых датчиков давления с архитектурой 2oo3 на входе. Логическая подсистема рассматриваемой системы представляет собой ПЭ систему с избыточностью с архитектурой 1oo2D и управляет одним закрывающим и одним дренажным клапанами, так как для обеспечения функции безопасности необходима работа как закрывающего, так и дренажного клапана. Архитектура всей системы представлена на рисунке В.14. Для этой системы оценим сначала функцию безопасности PFD_{SYS} при одногодичном периоде контрольных испытаний. Таблицы В.6 – В.8 являются фрагментами таблицы В.3 для соответствующих данных на рисунке В.14.

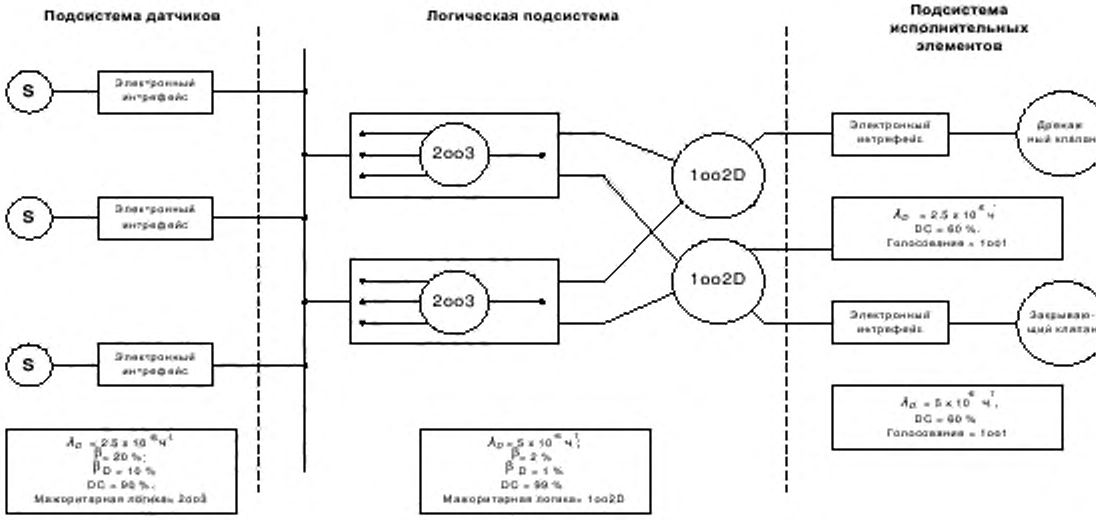


Рисунок В.13 – Архитектура системы рассматриваемого примера для режима низкой интенсивности запросов

Таблица В.6 – Средняя вероятность отказа по запросу для подсистемы датчиков в рассматриваемом примере для режима низкой интенсивности запросов (интервал контрольных испытаний равен одному году, а среднее время ремонта – 8 ч)

Архитектура	DC	λ_D = 2.5E-06		
		β = 2 % β_D = 1 %	β =10 % β_D=5 %	β = 20 % β_D=10 %
2oo3	0 %	6.8E-04	1.5E-03	2.5E-03
	60 %	1.6E-04	5.1 E-04	9.4E-04
	90 %	2.7E-05	1.2E-04	2.3E-04
	99 %	2.5E-06	1.2E-05	2.4E-05
Примечание - Настоящая таблица представляет собой фрагмент таблицы В.3.				

Таблица В.7 – Средняя вероятность отказа по запросу для логической подсистемы в примере для режима низкой интенсивности запросов (интервал контрольных испытаний равен одному году, а среднее время ремонта – 8 ч)

Архитектура	*DC*	λ_D =0.5E-05		
		β = 2 % β_D = 1 %	β =10 % β_D =5 %	β = 20 % β_D =10 %
1oo2D	0 %	1.1 E-03	2.7E-03	4.9E-03
	60 %	3.8E-04	1.1E-03	1.9E-03
	90 %	9.1E-05	2.7E-04	4.8E-04
	99 %	1.0E-05	2.8E-05	5.0E-05
Примечание - Настоящая таблица представляет собой фрагмент таблицы В.3.				

Таблица В.8 – Средняя вероятность отказа по запросу для подсистемы исполнительных элементов в примере для режима низкой интенсивности запросов (интервал контрольных испытаний равен одному году, а среднее время ремонта – 8 ч)

Архитектура	*DC*	λ_D =2.5E-06	λ_D =0.5E-05
1oo1	0 %	1.1 E-02	2.2E-02
	60 %	4.4E-03	8.8E-03
	90 %	1.1 E-03	2.2E-03
	99 %	1.3E-04	2.6E-04
Примечание - Настоящая таблица представляет собой фрагмент таблицы В.3.			

Данные, представленные в таблицах В.6 – В.8, позволяют получить следующие значения:

- для подсистемы датчиков:

$$PDF_S = 2{,}3 \times 10^{-4}$$

- для логической подсистемы:

$$PFD_L = 1{,}0 \times 10^{-5}$$

- для подсистемы исполнительных элементов:

$$PFD_{FE} = 4{,}4 \times 10^{-3} + 8{,}8 \times 10^{-3} = 1{,}3 \times 10^{-2}\ .$$

Следовательно, для функции безопасности:

$$PFD_{SYS} = 2.3 \times 10^{-4} + 1{,}0 \times 10^{-5} + 1{,}3 \times 10^{-2} = 1{,}3 \times 10^{-2}$$

- уровень полноты безопасности 1.

Для перевода системы на уровень полноты безопасности 2, выполняют одно из

следующих действий:

a) уменьшают интервал между контрольными проверками до 6 мес:

$$PDF_S = 1{,}1 \times 10^{-4}$$

$$PDF_L = 6{,}0 \times 10^{-6}$$

$$PDF_{FE} = 2{,}2 \times 10^{-3} + 4{,}4 \times 10^{-3} = 6{,}6 \times 10^{-3}$$

$$PDF_{SYS} = 6{,}7 \times 10^{-3}$$

- уровень полноты безопасности 2;

b) заменяют архитектуру 1oo1 закрывающего клапана, представляющего собой выходное устройство с низкой надежностью, на 1oo2, предполагая, что β = 10 % и β_D = 5 %:

$$PDF_S = 2{,}3 \times 10^{-4}$$

$$PFD_L = 1{,}0 \times 10^{-5}$$

$$PDF_{FE} = 4{,}4 \times 10^{-3} + 9{,}7 \times 10^{-4} = 5{,}4 \times 10^{-2}$$

$$PDF_{SYS} = 5{,}6 \times 10^{-2}$$

- уровень полноты безопасности 2.

B.3.2.5 Влияние неидеальных контрольных проверок

Отказы в системе безопасности, которые не обнаружены ни диагностическими тестами, ни контрольными проверками, могут быть обнаружены другими методами, реализуемыми в таких ситуациях, как появление опасного события, требующее вмешательства функции безопасности или во время капитального ремонта оборудования. Если отказы не будут обнаружены при помощи таких методов, следует предположить, что они останутся на весь срок службы оборудования. Обозначим обычный период между контрольными проверками T_1; долю отказов, обнаруженных контрольными проверками, обозначим как *PTC* (proof test coverage - охват контрольными проверками), а часть отказов, не обнаруженных контрольными проверками, как (1-*PTC*). Эти последние отказы, которые не могут быть обнаружены контрольными проверками, будут обнаружены только при запросе к системе, связанной с безопасностью, которые выполняются с интервалом T_2. Таким образом, период контрольных проверок T_1 и время между запросами T_2 управляют эффективностью простоя.

Далее приведен пример такой зависимости для архитектуры 1oo2, где T_2 – время между запросами к системе:

$$t_{GE} = \frac{\lambda_{DU}(PTC)}{\lambda_D}\left(\frac{T_1}{2} + MRT\right) + \frac{\lambda_{DU}(1 - PTC)}{\lambda_D}\left(\frac{T_2}{2} + MRT\right) + \frac{\lambda_{DD}}{\lambda_d} MTTR \ ,$$

$$t_{GE} = \frac{\lambda_{DU}(PTC)}{\lambda_D}\left(\frac{T_1}{3} + MRT\right) + \frac{\lambda_{DU}(1 - PTC)}{\lambda_d}\left(\frac{T_2}{3} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR,$$

$$PFD_G = 2((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU})^2 t_{CE} t_{GE} + \beta_D \lambda_{DD} MTTR + \beta\lambda_{DU}(PTC)\left(\frac{T_1}{2} + MRT\right) +$$

$$+\beta\lambda_{DU}(1 - PTC)\left(\frac{T_2}{2} + MRT\right).$$

Результаты для системы с архитектурой 1oo2 со 100 %-ными достоверными одногодичными контрольными проверками в сравнении с 90 %-ными достоверными контрольными проверками, где период запросов T_2 предполагается равным 10 годам, приведены в таблице В.9. В рассматриваемом примере расчеты проводились при следующих предположениях: интенсивность отказов 0.5×10^{-5} в час; β = 10 %; β_D = 5 %.

Таблица В.9 — Неидеальные контрольные испытания

Архитектура	*DC*	λ_D =0.5E-05	
		100 %-ные достоверные контрольные испытания	90 %-ные достоверные контрольные испытания
		β =10 % β_D = 5 %	β =10 % β_D = 5 %
1oo2	0 %	2.7E-03	6.0E-03
	60 %	9.7E-04	2.0E-03
	90 %	2.3E-04	4.4E-04
	99 %	2.4E-05	4.4E-05

В.3.3 Средняя интенсивность опасного отказа (для режима работы с высокой интенсивностью запросов или режима с непрерывным запросом)

В.3.3.1 Процедура расчетов

Метод определения вероятности отказа функции безопасности для Э/Э/ПЭ системы, связанной с безопасностью, работающей в режиме с высокой интенсивностью запросов или в режиме с непрерывным запросом, тот же, что и метод вычисления для режима с низкой интенсивностью запросов (см. В.2.1), за исключением того, что средняя вероятность отказа по запросу PFD_{SYS} заменяется на среднюю частоту опасного отказа в час PFH_{SYS}.

Общую вероятность опасного отказа функции безопасности для Э/Э/ПЭ системы, связанной с безопасностью, PFH_{SYS}, определяют вычислением интенсивностей опасных отказов для всех подсистем, совокупность которых обеспечивает функцию безопасности, и суммированием полученных значений. Так как рассматриваемые в

настоящем приложении вероятности малы, то используют формулу

$$PFH_{SYS} = PFH_S + PFH_L + PFH_{FE},$$

где PFH_{SYS} – средняя частота опасного отказа для функции безопасности Э/Э/ПЭ системы, связанной с безопасностью;

PFH_S - средняя частота опасного отказа для подсистемы датчиков;

PFH_L - средняя частота опасного отказа для логической подсистемы;

PFH_{FE} - средняя частота опасного отказа для подсистемы исполнительных элементов.

B.3.3.2 Архитектуры для режима работы высокой интенсивности запросов или в режиме с непрерывным запросом

Примечания

1 В настоящем пункте справедливые для нескольких архитектур формулы выводят там, где они встречаются впервые. См. также B.3.2.2.

2 Формулы настоящего пункта справедливы для предположений, перечисленных в B.3.1.

B.3.3.2.1 Архитектура 1oo1

На рисунках B.4 и B.5 представлены соответствующие структурные схемы. Для вычисления λ_D. t_{CE} , λ_{DU} и λ_{DD} используют те же формулы, что и в B.3.2.2.1.

$\lambda_d = \lambda_{DU} + \lambda_{DD}$.

$$t_{CE} = \frac{\lambda_{DU}}{\lambda_D}\left(\frac{T_1}{2} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR ,$$

$$\lambda_{DU} = \frac{\lambda}{2}(1 - DC) \; ; \; \lambda_{DD} = \frac{\lambda}{2} DC .$$

Если предположить, что система, связанная с безопасностью, при обнаружении любого отказа переводит УО в безопасное состояние, то для архитектуры 1oo1

$$PFH_G = \lambda_{DU} .$$

B. 3.3.2.2 Архитектура 1oo2

На рисунках B.6 и B.7 представлены соответствующие структурные схемы. Значение t_{CE} вычисляют по формуле, приведенной в B.3.3.2.1. Если подразумевается, что система безопасности переводит УО в безопасное состояние сразу после обнаружения отказа в обоих каналах и принимается консервативный подход, то применяется следующая формула

$$PFH_G = 2((1 - \beta_D)\lambda_{DD} + (1 - \beta)\lambda_{DU})(1 - \beta)\lambda_{DU} t_{CE} + \beta\lambda_{DU}.$$

B. 3.3.2.3 Архитектура 2oo2

Соответствующие структурные схемы представлены на рисунках B.8 и B.9. Если предположить, что при обнаружении любого отказа каждый канал переводится в безопасное состояние, то для архитектуры 2oo2

$$PFH_G = 2\lambda_{DU}.$$

B. 3.3.2.4 Архитектура 1oo2D

Соответствующие структурные схемы представлены на рисунках B.10 и B.11.

$$\lambda_{SO} = \frac{\lambda}{2} DC.$$

$$t_{CE}^{'} = \frac{\lambda_{DU}\left(\frac{T_1}{2} + MRT\right) + (\lambda_{DD} + \lambda_{SD})MTTR}{\lambda_{DU} + (\lambda_{DD} + \lambda_{SD})},$$

$$PFH_G = 2(1-\beta)\lambda_{DU}\left((1-\beta)\lambda_{DU} + (1-\beta_D)\lambda_{DD} + \lambda_{SD}\right)t_{CE}^{'} + 2(1-K)\lambda_{DD} + \beta\lambda_{DU}.$$

B. 3.3.2.5 Архитектура 2oo3

Соответствующие структурные схемы представлены на рисунках B.12 и B.13. Значение t_{CE} вычисляют по формуле, приведенной в B.3.3.2.1. Если подразумевается, что система безопасности переводит УО в безопасное состояние сразу после обнаружения отказа в любом из каналов и принимается консервативный подход, то применима следующая формула

$$PFH_G = 6\left((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\right)(1-\beta)\lambda_{DU}t_{CE} + \beta\lambda_{DU}.$$

B. 3.3.2.6 Архитектура 1oo3

Соответствующие структурные схемы представлены на рисунках B.12 и B.13. Значение t_{CE} вычисляют по формуле, приведенной в B.3.2.1. Если подразумевается, что система безопасности переводит УО в безопасное состояние сразу после обнаружения отказа в трех каналах и принимается консервативный подход, то применима следующая формула

$$PFH_G = 6\left((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\right)^2(1-\beta)\lambda_{DU}t_{CE}t_{GE} + \beta\lambda_{DU}.$$

В.3.3.3 Подробные таблицы для режима работы высокой интенсивности запросов и режима с непрерывным запросом

Таблица В.10 – Средняя частота опасных отказов (в режиме работы высокой интенсивности запросов и с непрерывным запросом) для одномесячного интервала между контрольными проверками и среднего времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	5.0E-08			2.5E-07			5.0E-07		
	60 %	2.0E-08			1.0E-07			2.0E-07		
	90 %	5.0E-09			2.5E-08			5.0E-08		
	99 %	5.0E-10			2.5E-09			5.0E-09		
1oo2	0 %	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.0E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
2oo2 (см. примечание 2)	0 %	1.0E-07			5.0E-07			1.0E-06		
	60 %	4.0E-08			2.0E-07			4.0E-07		
	90 %	1.0E-08			5.0E-08			1.0E-07		
	99 %	1.0E-09			5.0E-09			1.0E-08		
1oo2D (см. примечание 3)	0 %	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07
	60 %	1.6E-09	3.2E-09	5.2E-09	8.0E-09	1.6E-08	2.6E-08	1.6E-08	3.2E-08	5.2E-08
	90 %	1.9E-09	2.3E-09	2.8E-09	9.5E-09	1.2E-08	1.4E-08	1.9E-08	2.3E-08	2.8E-08
	99 %	2.0E-09	2.0E-09	2.1E-09	1.0E-08	1.0E-08	1.0E-08	2.0E-08	2.0E-08	2.1E-08
2oo3	0 %	1.0E-09	5.0E-09	1.0E-08	5.1E-09	2.5E-08	5.0E-08	1.1 E-08	5.0E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.1E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
1oo3	0 %	1.0E-09	5.0E-09	1.0E-08	5.1E-09	2.5E-08	5.0E-08	1.1 E-08	5.0E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.0E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09

Окончание таблицы В.10

Архитектура	*DC*	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	2.5E-06			5.0E-06			2.5E-05		
	60 %	1.0E-06			2.0E-06			1.0E-05		
	90 %	2.5E-07			5.0E-07			2.5E-06		
	99 %	2.5E-08			5.0E-08			2.5E-07		
1oo2	0 %	5.4E-08	2.5E-07	5.0E-07	1.2E-07	5.2E-07	1.0E-06	9.5E-07	2.9E-06	5.3E-06
	60 %	2.1E-08	1.0E-07	2.0E-07	4.3E-08	2.0E-07	4.0E-07	2.7E-07	1.1E-06	2.1E-06
	90 %	5.1E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07	5.5E-08	2.5E-07	5.0E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.1E-09	2.5E-08	5.0E-08
2oo2 (см. примечание 2)	0 %	5.0E-06			1.0E-05			5.0E-05		
	60 %	2.0E-06			4.0E-06			2.0E-05		
	90 %	5.0E-07			1.0E-06			5.0E-06		
	99 %	5.0E-08			1.0E-07			5.0E-07		
1oo2D (см. примечание 3)	0 %	5.4E-08	2.5E-07	5.0E-07	1.2E-07	5.2E-07	1.0E-06	9.5E-07	2.9E-06	5.3E-06
	60 %	8.1E-08	1.6E-07	2.6E-07	1.6E-07	3.2E-07	5.2E-07	8.7E-07	1.7E-06	2.7E-06
	90 %	9.5E-08	1.2E-07	1.4E-07	1.9E-07	2.3E-07	2.8E-07	9.6E-07	1.2E-06	1.4E-06
	99 %	1.0E-07	1.0E-07	1.0E-07	2.0E-07	2.0E-07	2.1E-07	1.0E-06	1.0E-06	1.0E-06
2oo3	0 %	6.3E-08	2.6E-07	5.1 E-07	1.5E-07	5.5E-07	1.0E-06	1.8E-06	3.6E-06	5.9E-06
	60 %	2.2E-08	1.0E-07	2.0E-07	4.9E-08	2.1E-07	4.1E-07	4.2E-07	1.2E-06	2.2E-06
	90 %	5.2E-09	2.5E-08	5.0E-08	1.1E-08	5.1E-08	1.0E-07	6.6E-08	2.6E-07	5.1E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.4E-09	2.5E-08	5.0E-08
1oo3	0 %	5.0E-08	2.5E-07	5.0E-07	1.0E-07	5.0E-07	1.0E-06	5.1E-07	2.5E-06	5.0E-06
	60 %	2.0E-08	1.0E-07	2.0E-07	4.0E-08	2.0E-07	4.0E-07	2.0E-07	1.0E-06	2.0E-06
	90 %	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07	5.0E-08	2.5E-07	5.0E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08

Примечания

1 В настоящей таблице приведены примеры значений PFH_G, рассчитанные по формулам в соответствии с В.3.3 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной голосующей группы, то PFH_G эквивалентна PFH_S, PFH_L, или PFH_{FE} соответственно (см. В.3.3.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

Таблица В.11 – Средняя частота опасных отказов (в режиме работы высокой интенсивности запросов и с непрерывным запросом) для трехмесячного интервала между контрольными проверками и среднего времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	5.0E-08			2.5E-07			5.0E-07		
	60 %	2.0E-08			1.0E-07			2.0E-07		
	90 %	5.0E-09			2.5E-08			5.0E-08		
	99 %	5.0E-10			2.5E-09			5.0E-09		
1oo2	0 %	1.0E-09	5.0E-09	1.0E-08	5.1E-09	2.5E-08	5.0E-08	1.1E-08	5.0E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.1E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
2oo2 (см. примечание 2)	0 %	1.0E-07			5.0E-07			1.0E-06		
	60 %	4.0E-08			2.0E-07			4.0E-07		
	90 %	1.0E-08			5.0E-08			1.0E-07		
	99 %	1.0E-09			5.0E-09			1.0E-08		
1oo2D (см. примечание 3)	0 %	1.0E-09	5.0E-09	1.0E-08	5.1 E-09	2.5E-08	5.0E-08	1.1E-08	5.0E-08	1.0E-07
	60 %	1.6E-09	3.2E-09	5.2E-09	8.0E-09	1.6E-08	2.6E-08	1.6E-08	3.2E-08	5.2E-08
	90 %	1.9E-09	2.3E-09	2.8E-09	9.5E-09	1.2E-08	1.4E-08	1.9E-08	2.3E-08	2.8E-08
	99 %	2.0E-09	2.0E-09	2.1E-09	1.0E-08	1.0E-08	1.0E-08	2.0E-08	2.0E-08	2.1E-08
2oo3	0 %	1.0E-09	5.0E-09	1.0E-08	5.4E-09	2.5E-08	5.0E-08	1.2E-08	5.1E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.1E-09	1.0E-08	2.0E-08	4.3E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
1oo3	0 %	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.0E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09

Окончание таблицы В.11

Архитектура	*DC*	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	2.5E-06			5.0E-06			2.5E-05		
	60 %	1.0E-06			2.0E-06			1.0E-05		
	90 %	2.5E-07			5.0E-07			2.5E-06		
	99 %	2.5E-08			5.0E-08			2.5E-07		
1oo2	0 %	6.3E-08	2.6E-07	5.1E-07	1.5E-07	5.4E-07	1.0E-06	1.8E-06	3.6E-06	5.9E-06
	60 %	2.2E-08	1.0E-07	2.0E-07	4.9E-08	2.1E-07	4.1E-07	4.2E-07	1.2E-06	2.2E-06
	90 %	5.1E-09	2.5E-08	5.0E-08	1.1E-08	5.0E-08	1.0E-07	6.4E-08	2.6E-07	5.1E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.2E-09	2.5E-08	5.0E-08
2oo2 (см. примечание 2)	0 %	5.0E-06			1.0E-05			5.0E -05		
	60 %	2.0E-06			4.0E-06			2.0E -05		
	90 %	5.0E-07			1.0E-06			5.0E-06		
	99 %	5.0E-08			1.0E-07			5.0E-07		
1oo2D (см. примечание 3)	0 %	6.3E-08	2.6E-07	5.1 E-07	1.5E-07	5.4E-07	1.0E-06	1.8E-06	3.6E-06	5.9E-06
	60 %	8.2E-08	1.6E-07	2.6E-07	1.7E-07	3.3E-07	5.3E-07	1.0E-06	1.8E-06	2.8E-06
	90 %	9.5E-08	1.2E-07	1.4E-07	1.9E-07	2.3E-07	2.8E-07	9.6E-07	1.2E-06	1.4E-06
	99 %	1.0E-07	1.0E-07	1.0E-07	2.0E-07	2.0E-07	2.1E-07	1.0E-06	1.0E-06	1.0E-06
2oo3	0 %	9.0E-08	2.8E-07	5.3E-07	2.6E-07	6.3E-07	1.1E-06	4.5E-06	5.9E-06	7.6E-06
	60 %	2.6E-08	1.1E-07	2.0E-07	6.6E-08	2.2E-07	4.2E-07	8.5E-07	1.6E-06	2.5E-06
	90 %	5.4E-09	2.5E-08	5.0E-08	1.2E-08	5.1E-08	1.0E-07	9.3E-08	2.9E-07	5.3E-07
	99 %	5.1E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.7E-09	2.6E-08	5.1E-08
1oo3	0 %	5.0E-08	2.5E-07	5.0E-07	1.0E-07	5.0E-07	1.0E-06	5.1E-07	2.5E-06	5.0E-06
	60 %	2.0E-08	1.0E-07	2.0E-07	4.0E-08	2.0E-07	4.0E-07	2.0E-07	1.0E-06	2.0E-06
	90 %	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07	5.0E-08	2.5E-07	5.0E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08

Примечания

1 В настоящей таблице приведены примеры значений PFH_G, рассчитанные по формулам в соответствии с В.3.3 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной голосующей группы, то PFH_G эквивалентна PFH_S, PFH_L. или PFH_{FE} соответственно (см. В.3.3.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

Таблица В.12 – Средняя частота опасных отказов (в режиме работы высокой интенсивности запросов и с непрерывным запросом) для шестимесячного интервала между контрольными проверками и для среднего времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	5.0E-08			2.5E-07			5.0E-07		
	60 %	2.0E-08			1.0E-07			2.0E-07		
	90 %	5.0E-09			2.5E-08			5.0E-08		
	99 %	5.0E-10			2.5E-09			5.0E-09		
1oo2	0 %	1.0E-09	5.0E-09	1.0E-08	5.3E-09	2.5E-08	5.0E-08	1.1E-08	5.1E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.2E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
2oo2 (см. примечание 2)	0 %	1.0E-07			5.0E-07			1.0E-06		
	60 %	4.0E-08			2.0E-07			4.0E-07		
	90 %	1.0E-08			5.0E-08			1.0E-07		
	99 %	1.0E-09			5.0E-09			1.0E-08		
1oo2D (см. примечание 3)	0 %	1.0E-09	5.0E-09	1.0E-08	5.3E-09	2.5E-08	5.0E-08	1.1E-08	5.1E-08	1.0E-07
	60 %	1.6E-09	3.2E-09	5.2E-09	8.0E-09	1.6E-08	2.6E-08	1.6E-08	3.2E-08	5.2E-08
	90 %	1.9E-09	2.3E-09	2.8E-09	9.5E-09	1.2E-08	1.4E-08	1.9E-08	2.3E-08	2.8E-08
	99 %	2.0E-09	2.0E-09	2.1E-09	1.0E-08	1.0E-08	1.0E-08	2.0E-08	2.0E-08	2.1E-08
2oo3	0 %	1.0E-09	5.0E-09	1.0E-08	5.8E-09	2.6E-08	5.1E-08	1.3E-08	5.3E-08	1.0E-07
	60 %	4.1E-10	2.0E-09	4.0E-09	2.1E-09	1.0E-08	2.0E-08	4.5E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.1E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
1oo3	0 %	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.0E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09

Окончание таблицы В.12

Архитектура	*DC*	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	2.5E-06			5.0E-06			2.5E-05		
	60 %	1.0E-06			2.0E-06			1.0E-05		
	90 %	2.5E-07			5.0E-07			2.5E-06		
	99 %	2.5E-08			5.0E-08			2.5E-07		
1oo2	0 %	7.6E-08	2.7E-07	5.2E-07	2.1 E-07	5.9E-07	1.1E-06	3.1 E-06	4.7E-06	6.8E-06
	60 %	2.4E-08	1.0E-07	2.0E-07	5.7E-08	2.1E-07	4.1E-07	6.3E-07	1.4E-06	2.3E-06
	90 %	5.3E-09	2.5E-08	5.0E-08	1.1E-08	5.1E-08	1.0E-07	7.8E-08	2.7E-07	5.2E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.4E-09	2.5E-08	5.0E-08
2oo2 (см. примечание 2)	0 %	5.0E-06			1.0E-05			5.0E-05		
	60 %	2.0E-06			4.0E-06			2.0E-05		
	90 %	5.0E-07			1.0E-06			5.0E-06		
	99 %	5.0E-08			1.0E-07			5.0E-07		
1oo2D (см. примечание 3)	0 %	7.6E-08	2.7E-07	5.2E-07	2.1E-07	5.9E-07	1.1 E-06	3.1 E-06	4.7E-06	6.8E-06
	60 %	8.4E-08	1.6E-07	2.6E-07	1.8E-07	3.3E-07	5.3E-07	1.2E-06	2.0E-06	2.9E-06
	90 %	9.5E-08	1.2E-07	1.4E-07	1.9E-07	2.3E-07	2.8E-07	9.8E-07	1.2E-06	1.4E-06
	99 %	1.0E-07	1.0E-07	1.0E-07	2.0E-07	2.0E-07	2.1E-07	1.0E-06	1.0E-06	1.0E-06
2oo3	0 %	1.3E-07	3.2E-07	5.5E-07	4.2E-07	7.7E-07	1.2E-06	8.4E-06	9.2E-06	1.0E-05
	60 %	3.3E-08	1.1E-07	2.1E-07	9.1E-08	2.4E-07	4.4E-07	1.5E-06	2.1E-06	2.9E-06
	90 %	5.8E-09	2.6E-08	5.1E-08	1.3E-08	5.3E-08	1.0E-07	1.3E-07	3.2E-07	5.6E-07
	99 %	5.1E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	6.1E-09	2.6E-08	5.1E-08
1oo3	0 %	5.0E-08	2.5E-07	5.0E-07	1.0E-07	5.0E-07	1.0E-06	7.1E-07	2.7E-06	5.1E-06
	60 %	2.0E-08	1.0E-07	2.0E-07	4.0E-08	2.0E-07	4.0E-07	2.1E-07	1.0E-06	2.0E-06
	90 %	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07	5.0E-08	2.5E-07	5.0E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08

Примечания

1 В настоящей таблице приведены примеры значений PFH_G, рассчитанные по формулам в соответствии с В.3.3 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной голосующей группы, то PFH_G эквивалентна PFH_S, PFH_L. или PFH_{FE} соответственно (см. В.3.3.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

Таблица В.13 – Средняя частота опасных отказов (в режиме работы высокой интенсивности запросов и с непрерывным запросом) для одногодичного интервала между контрольными проверками и для среднего времени ремонта 8 ч

Архитектура	*DC*	λ_D = 0.5E-07			λ_D = 2.5E-07			λ_D = 0.5E-06		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	5.0E-08			2.5E-07			5.0E-07		
	60 %	2.0E-08			1.0E-07			2.0E-07		
	90 %	5.0E-09			2.5E-08			5.0E-08		
	99 %	5.0E-10			2.5E-09			5.0E-09		
1oo2	0 %	1.0E-09	5.0E-09	1.0E-08	5.5E-09	2.5E-08	5.0E-08	1.2E-08	5.2E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.1E-09	1.0E-08	2.0E-08	4.3E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.1E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
2oo2 (см. примечание 2)	0 %	1.0E-07			5.0E-07			1.0E-06		
	60 %	4.0E-08			2.0E-07			4.0E-07		
	90 %	1.0E-08			5.0E-08			1.0E-07		
	99 %	1.0E-09			5.0E-09			1.0E-08		
1oo2D (см. примечание 3)	0 %	1.0E-09	5.0E-09	1.0E-08	5.5E-09	2.5E-08	5.0E-08	1.2E-08	5.2E-08	1.0E-07
	60 %	1.6E-09	3.2E-09	5.2E-09	8.1E-09	1.6E-08	2.6E-08	1.6E-08	3.2E-08	5.2E-08
	90 %	1.9E-09	2.3E-09	2.8E-09	9.5E-09	1.2E-08	1.4E-08	1.9E-08	2.3E-08	2.8E-08
	99 %	2.0E-09	2.0E-09	2.1E-09	1.0E-08	1.0E-08	1.0E-08	2.0E-08	2.0E-08	2.1E-08
2oo3	0 %	1.1E-09	5.1E-09	1.0E-08	6.6E-09	2.6E-08	5.1E-08	1.6E-08	5.5E-08	1.0E-07
	60 %	4.1E-10	2.0E-09	4.0E-09	2.3E-09	1.0E-08	2.0E-08	5.0E-09	2.1E-08	4.1E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.2E-10	2.5E-09	5.0E-09	1.1E-09	5.1E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09
1oo3	0 %	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07
	60 %	4.0E-10	2.0E-09	4.0E-09	2.0E-09	1.0E-08	2.0E-08	4.0E-09	2.0E-08	4.0E-08
	90 %	1.0E-10	5.0E-10	1.0E-09	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08
	99 %	1.0E-11	5.0E-11	1.0E-10	5.0E-11	2.5E-10	5.0E-10	1.0E-10	5.0E-10	1.0E-09

Окончание таблицы В.13

Архитектура	*DC*	λ_D = 2.5E-06			λ_D = 0.5E-05			λ_D = 2.5E-05		
		β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %	β=2 % β_D=1 %	β=10 % β_D =5 %	β=20 % β_D=10 %
1oo1 (см. примечание 2)	0 %	2.5E-06			5.0E-06			2.5E-05		
	60 %	1.0E-06			2.0E-06			1.0E-05		
	90 %	2.5E-07			5.0E-07			2.5E-06		
	99 %	2.5E-08			5.0E-08			2.5E-07		
1oo2	0 %	1.0E-07	2.9E-07	5.4E-07	3.1E-07	6.8E-07	1.1E-06	5.8E-06	6.9E-06	8.5E-06
	60 %	2.9E-08	1.1E-07	2.1E-07	7.4E-08	2.3E-07	4.2E-07	1.1E-06	1.7E-06	2.6E-06
	90 %	5.5E-09	2.5E-08	5.0E-08	1.2E-08	5.2E-08	1.0E-07	1.0E-07	3.0E-07	5.4E-07
	99 %	5.1E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.6E-09	2.6E-08	5.0E-08
2oo2 (см. примечание 2)	0 %	5.0E-06			1.0E-05			5.0E-05		
	60 %	2.0E-06			4.0E-06			2.0E-05		
	90 %	5.0E-07			1.0E-06			5.0E-06		
	99 %	5.0E-08			1.0E-07			5.0E-07		
1oo2D (см. примечание 3)	0 %	1.0E-07	2.9E-07	5.4E-07	3.1E-07	6.8E-07	1.1E-06	5.8E-06	6.9E-06	8.5E-06
	60 %	8.9E-08	1.7E-07	2.7E-07	1.9E-07	3.5E-07	5.4E-07	1.7E-06	2.3E-06	3.2E-06
	90 %	9.6E-08	1.2E-07	1.4E-07	1.9E-07	2.3E-07	2.8E-07	1.0E-06	1.2E-06	1.4E-06
	99 %	1.0E-07	1.0E-07	1.0E-07	2.0E-07	2.0E-07	2.1E-07	1.0E-06	1.0E-06	1.0E-06
2oo3	0 %	2.1 E-07	3.8E-07	6.1E-07	7.3E-07	1.0E-06	1.4E-06	1.6E -05	1.6E-05	1.6E-05
	60 %	4.6E-08	1.2E-07	2.2E-07	1.4E-07	2.9E-07	4.7E-07	2.8E-06	3.2E-06	3.8E-06
	90 %	6.6E-09	2.6E-08	5.1E-08	1.6E-08	5.6E-08	1.0E-07	2.1E-07	3.9E-07	6.2E-07
	99 %	5.2E-10	2.5E-09	5.0E-09	1.1E-09	5.1E-09	1.0E-08	6.9E-09	2.7E-08	5.1E-08
1oo3	0 %	5.1E-08	2.5E-07	5.0E-07	1.1E-07	5.1E-07	1.0E-06	1.4E-06	3.2E-06	5.5E-06
	60 %	2.0E-08	1.0E-07	2.0E-07	4.0E-08	2.0E-07	4.0E-07	2.6E-07	1.0E-06	2.0E-06
	90 %	5.0E-09	2.5E-08	5.0E-08	1.0E-08	5.0E-08	1.0E-07	5.1E-08	2.5E-07	5.0E-07
	99 %	5.0E-10	2.5E-09	5.0E-09	1.0E-09	5.0E-09	1.0E-08	5.0E-09	2.5E-08	5.0E-08

Примечания

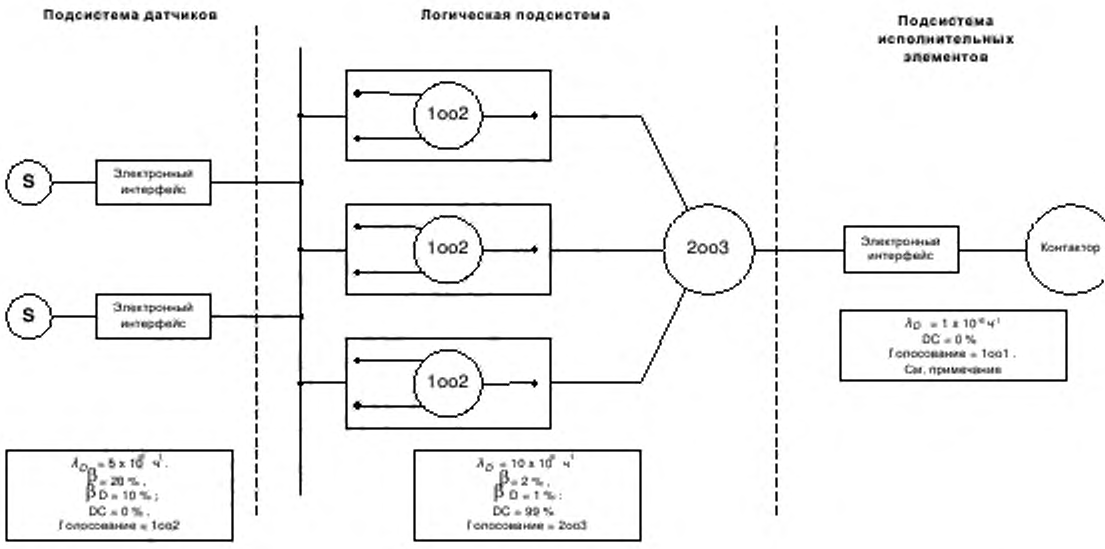
1 В настоящей таблице приведены примеры значений PFH_G, рассчитанные по формулам в соответствии с В.3.3 и с учетом предположений, перечисленных в В.3.1. Если подсистема датчиков, логическая подсистема или подсистема исполнительных элементов входит в состав только одной голосующей группы, то PFH_G эквивалентна PFH_S, PFH_L, или PFH_{FE} соответственно (см. В.3.3.1).

2 В настоящей таблице предполагается, что $\beta = 2 \times \beta_D$. Для архитектур 1oo1 и 2oo2 значения β и β_D не влияют на среднюю вероятность отказа.

3 Интенсивность безопасных отказов принимается равной интенсивности опасных отказов и K=0.98

В.3.3.4. Пример режима работы высокой интенсивности запросов или в режиме с непрерывным запросом

Рассмотрим функцию безопасности, для реализации которой нужна система УПБ2. Пусть первоначальный вариант архитектуры всей системы, построенный на основе предыдущего опыта, включает одну группу из двух датчиков с архитектурой 1oo2 на входе. Логическая подсистема рассматриваемой системы представляет собой ПЭ систему с избыточностью с архитектурой 2oo3 и управляет одним закрывающим контактором. Архитектура описанной системы представлена на рисунке В.15. Для этой системы оценим значение при шестимесячном интервале между контрольными проверками. Таблицы В.14 – В.16 являются фрагментами таблицы В.12 для соответствующих данных на рисунке В.15.



Примечание – Доля безопасных отказов для подсистемы исполнительных элементов превышает 60 %.

Рисунок В.15 – Архитектура системы рассматриваемого примера для режима высокой интенсивности запросов или режима с непрерывным запросом

Таблица В.14 – Средняя частота опасных отказов для подсистемы датчиков в рассматриваемом примере режима работы высокой интенсивности запросов или режима с непрерывным запросом (шестимесячный интервал контрольных проверок и среднее время ремонта 8 ч)

Архитектура	*DC*	λ_D = 2.5E-06		
		β=2 % β_D= 1 %	β =10 % β_D=5 %	β = 20 % β_D=10 %
1oo2	0 %	7.6E-08	2.7E-07	5.2E-07
	60 %	2.4E-08	1.0E-07	2.0E-07
	90 %	5.3E-09	2.5E-08	5.0E-08
	99 %	5.0E-10	2.5E-09	5.0E-09
Примечание – Настоящая таблица представляет собой фрагмент таблицы В.12.				

Таблица В.15 – Средняя частота опасных отказов для логической подсистемы в рассматриваемом примере режима работы высокой интенсивности запросов или режима с непрерывным запросом (шестимесячный интервал контрольных проверок и среднее время ремонта 8 ч)

Архитектура	*DC*	λ_D = 0.5E-05		
		β=2 % β_D= 1 %	β =10 % β_D=5 %	β = 20 % β_D=10 %
2oo3	0 %	4.2E-07	7.7E-07	1.2E-06
	60 %	9.1E-08	2.4E-07	4.4E-07
	90 %	1.3E-08	5.3E-08	1.0E-07
	99 %	1.0E-09	5.0E-09	1.0E-08
Примечание – Настоящая таблица представляет собой фрагмент таблицы В.12.				

Таблица В.16 – Средняя частота опасных отказов для подсистемы исполнительных элементов в рассматриваемом примере режима работы высокой интенсивности запросов или режима с непрерывным запросом (шестимесячный интервал контрольных испытаний и среднее время ремонта 8 ч)

Архитектура	*DC*	λ_D = 0.5E-06
1oo1	0 %	5.0E-07
	60 %	2.0E-07
	90 %	5.0E-08
	99 %	5.0E-09
Примечание – Настоящая таблица представляет собой фрагмент таблицы В.12.		

Данные таблиц В.14 – В.16 позволяют получить следующие значения:

- для подсистемы датчиков –

$PFH_S = 5{,}2 \times 10^{-7}$ / h ;

- для логической подсистемы –

$$PFH_L = 1{,}0 \times \frac{10^{-9}}{h} ;$$

- для подсистемы исполнительных элементов –

$PFH_{FE} = 5{,}0 \times 10^{-7}$ / h ;

следовательно, для функции безопасности –

$$PFH_{SYS} = 5{,}2 \times 10^{-7} + 1{,}0 \times 10^{-9} + 5{,}0 \times 10^{-7} = 1{,}02 \times \frac{10^{-6}}{h}$$

- уровень полноты безопасности 1.

Для перевода системы на уровень полноты безопасности 2, выполняют одно из следующих действий:

a) изменяют тип и способ установки входного датчика для улучшения защиты от отказа по общей причине. Таким образом, снижая значение β от 20 % до 10 %, а β_D от 10 % до 5 %, получаем:

$$PFH_S = 2{,}7 \times \frac{10^{-7}}{h} ,$$

$$PFH_L = 1{,}0 \times \frac{10^{-9}}{h} ,$$

$$PFH_{FE} = 5{,}0 \times \frac{10^{-7}}{h} ,$$

$$PFH_{SYS} = 7{,}7 \times \frac{10^{-7}}{h}$$

- уровень полноты безопасности 2;

b) заменяют единственное выходное устройство двумя устройствами с архитектурой 1oo2 (β = 10 % и β_D = 5 %):

$$PFH_S = 5{,}2 \times \frac{10^{-7}}{h} ,$$

$$PFH_L = 1{,}0 \times \frac{10^{-9}}{h} ,$$

$$PFH_{FE} = 5{,}1 \times \frac{10^{-8}}{h} ,$$

$$PFH_{SYS} = 5{,}7 \times \frac{10^{-7}}{h}$$

- уровень полноты безопасности 2.

B.4 Логический подход

B.4.1 Общие положения

Логический подход представляют собой методы, использующие логические функции, которые связывают отказы отдельных компонентов с общим отказом системы. Основными логическими моделями, использующимися в надежности, являются блок-схемы надежности, деревья отказов, деревья событий и причинно-следственные диаграммы. В настоящем стандарте рассматриваются только первые два метода. Целью всех этих методов является представление логической структуры системы. Тем не менее в моделях этих методов не учитывается ее поведение во времени. Поэтому при проведении расчетов необходимо проявлять осторожность при рассмотрении характеристик поведения системы (например, зависимых от времени характеристик типа периодических контрольных проверок). Первым шагом для применения логических моделей является отделение графического представления системы от вычислений. Это было описано в предыдущем разделе, где блок-схема надежности используется для моделирования структуры системы, а расчеты на основе моделей Маркова - для оценки *PDF* или *PFH*. Далее будут рассматриваться вероятностные расчеты для методов блок-схемы надежности и дерева отказов.

Данный подход ограничен тем, что поведение компонентов считается достаточно независимым друг от друга.

B.4.2 Модель блок-схемы надежности

Ранее было рассмотрено множество примеров блок-схемы надежности, например, на рисунке B.1 представлена полная система безопасности, состоящая из трех сенсоров (*A, B, C*), работающих по схеме 1oo3, одного логического решающего усройства (*D*) и двух исполнительных элементов (*E, F*), работающих по схеме 1oo2.

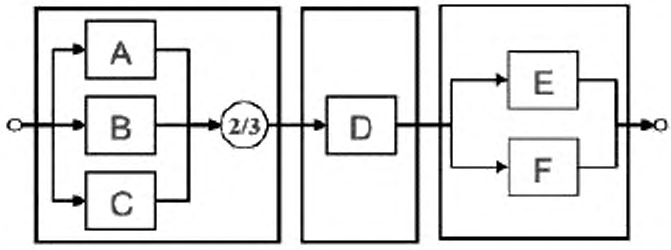


Рисунок B.16 – Блок-схема надежности простой полной системы безопасности с датчиками, организованными по схеме 2oo3

На рисунке B.16 представлена простая система безопасности с датчиками,

работающими по схеме голосования 2oo3. Основное удобство такого графического представления объясняется тремя позициями: оно очень близко к физической структуре изучаемой системы, широко используется в среде инженеров и оно наглядно, что удобно для обсуждения.

Основной недостаток блок-схемы надежности состоит в том, что этот метод в большей степени является методом представления, чем методом анализа. См. МЭК 61508-7, п. С.6.4 и [2].

В.4.3 Модель дерева отказов

Деревья отказов имеют те же свойства, что и блок-схемы надежности, но в дополнение ко всему они предоставляют эффективный дедуктивный (сверху вниз) метод анализа, помогающий инженерам по надежности разрабатывать модели шаг за шагом от события верхнего уровня (нежелаемого или недопустимого) к отказам отдельных компонентов.

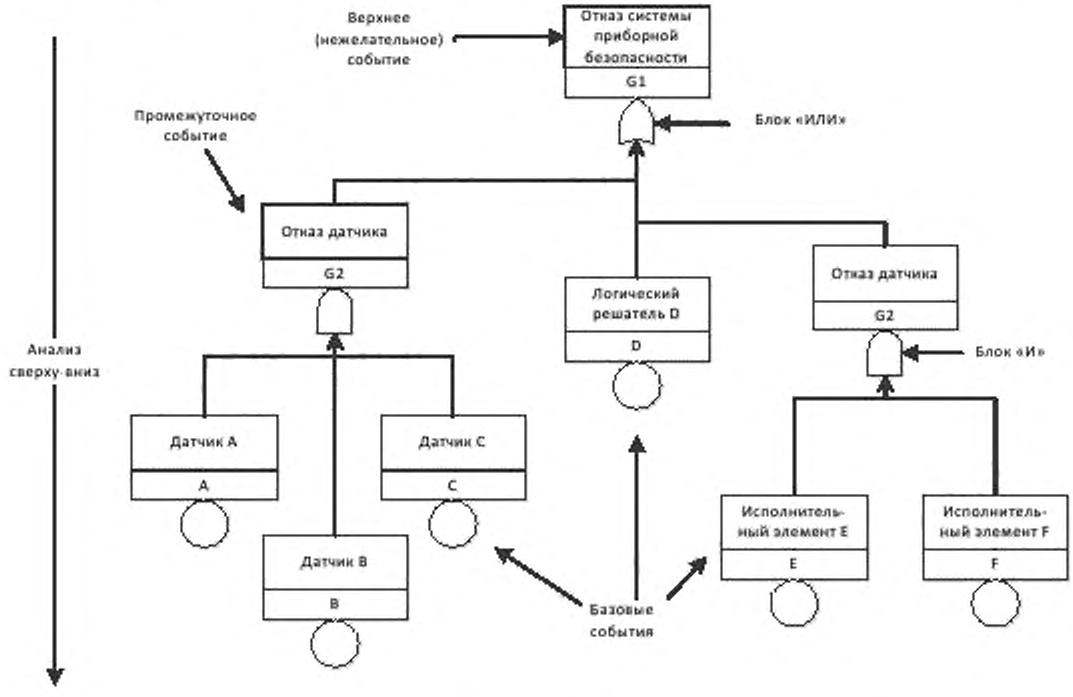


Рисунок В.17 – Простое дерево отказов, эквивалентное блок-схеме, предоставленной на рис. В.1

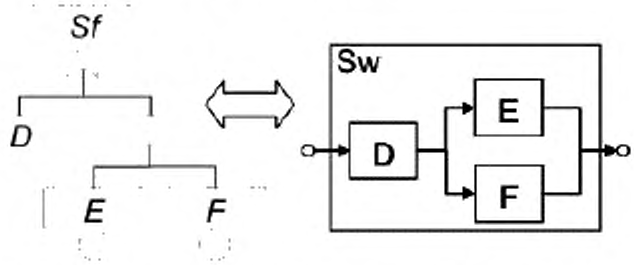
Рисунок В.17 показывает дерево отказов, которое является идеальным аналогом блок-схемы надежности, предоставленной на рис. В.1, но с указанными сверху вниз шагами анализа (например, отказ Э/Э/ПЭ системы, связанной с безопасностью => отказ

датчика => отказ датчика А). В дереве отказов элементы, работающие последовательно, соединены оператором «ИЛИ», а элементы, работающие параллельно (реализующие резервирование), – оператором «И». См. МЭК 61508-7, пункты В.6.6.5 и В.6.6.9 и [4].

В.4.4 Расчет PFD

В.4.4.1 Общие положения

Блок-схема надежности и дерево отказов представляют одно и то же и расчеты могут быть выполнены похожим способом. Рисунок 18 показывает небольшое сходство методов дерева отказов и блок-схемы надежности, которое будет использовано для демонстрации основных принципов вычислений.



Примечание – На рисунке курсивом обозначены отказавшие элементы, не курсивом – работающие.

Рисунок В.18 – Эквивалентность дерева отказов и блок-схемы надежности

Дерево отказов описывается логической функцией $Sf = D \cup (E \cap F)$, где *Sf* – это отказ системы, а *D*, *E* и *F* – отказы отдельных компонент. Блок-схема надежности

$$\mathbf{Sw} = \mathbf{D} \cap \mathbf{E} \cup$$

описывается логической функцией , где Sw – это правильно функционирующая система, а D, E и F – правильно функционирующие отдельные компоненты. Тогда *Sf* = НЕ Sw, а *Sf* и Sw представляют абсолютно идентичную информацию (т.е. являются дуальными (двойственными) логическими функциями).

Главное предназначение дерева отказов и блок-схемы надежности состоит в определении комбинаций отказов разных компонент, ведущих к общему отказу системы. Они также называются минимальными сечениями, потому что указывают, где «разрезается» блок-схема надежности, в результате чего сигнал, поданый на вход, не достигнет выхода. В данном случае имеем два вида сечений: одиночный отказ (*D*) и двойной отказ (*E*, *F*).

Применяя методы теории вероятности к логическим функциям, можно непосредственно вычислить вероятность отказа рассматриваемой системы P_{Sf}

$$P_{Sf} = P(D) + P(E \cap F) - P(D \cap E \cap F).$$

Если компоненты системы независимы, то эта формула имеет вид:

$$P_{Sf} = P_D + P_E P_F - P_D P_E P_F ,$$

где P_i – отказавший *i*-й компонент.

Данная формула является независимой от времени и отражает только логическую структуру системы.

Таким образом и блок-схема надежности и дерево отказов являются в основе своей статическими, т.е. моделями, независимыми от времени.

Тем не менее, если вероятность отказа каждого отдельного компонента в момент времени *t* не зависит от того, что происходит с другим компонентом в интервале [0, t], то указанная выше формула остается правильной в любой момент времени и мы можем написать:

$$P_{Sf}(t) = P_D(t) + P_E(t)P_F(t) - P_D(t)P_E(t)P_F(t).$$

Аналитик должен проверить, применимы ли требуемые приближения и, наконец, можно получить неготовность системы $U_{Sf}(t)$ в конкретный момент времени *t*:

$$U_{Sf}(t) = U_D(t) + U_E(t)U_F(t) - U_D(t)U_E(t)U_F(t).$$

Из этого можно сделать вывод, что деревья отказов и блок-схемы надежности позволяют вычислить мгновенную неготовность *USf(t)* Э/Э/ПЭ системы, связанной с безопасностью, и в соответствии с В.2.2 далее можно вычислить:

$$PFD_{avg}(T) = \frac{1}{T} MDT(T) = \frac{1}{T}\int_0^T U_{Sf}(t)dt .$$

Данный подход может быть применен и для минимальных сечений:

- одиночный отказ (D): $PDF^D(\tau) = \frac{1}{\tau}\int_0^\tau \lambda_D t.dt = \lambda_D \tau/2$;
- двойной отказ (E, F): $PDF^{EF}(\tau) = \frac{1}{\tau}\int_0^\tau \lambda_E \lambda_F .t^2 dt = \lambda_E \lambda_F \tau^2/3$.

В.4.4.2 Вычисления, выполняемые для реализации методов дерево отказов или блок-схема надежности

Описанная выше формула $U_{Sf}(t) = U_D(t) + U_E(t)U_F(t) - U_D(t)U_E(t)U_F(t)$ является частным видом так называемой формулы Пуанкаре. Более общая формула, когда $Sf = \bigcup_i C_i$, где (C_i) представляет собой минимальные сечения системы, имеет вид:

$$P\left(\bigcup_{i-1}^{n} C_i\right) = \sum_{j=1}^{n} P(C_j) - \sum_{j=1}^{n}\sum_{i=1}^{j-1} P(C_{j \cap C_i}) + \sum_{j=3}^{n}\sum_{i=2}^{j-1}\sum_{k=1}^{j-1} P\left(C_{j \cap C_{i \cap C_k}}\right) - \ldots$$

С увеличением количества отдельных компонент число минимальных сечений растет экспоненциально. В этом случае формула Пуанкаре приводит к комбинаторному взрыву числа вычисляемых элементов, что вручную выполнить невозможно. К счастью, эта проблема анализировалась в течение последних сорока лет и были созданы многочисленные алгоритмы для выполнения подобных расчетов. На сегодня наиболее эффективные разработки основаны на так называемой бинарной диаграмме решений (Binary Decision Diagrams, BDD), которая получена из развитого Шенноном разложения логической функции.

Множество коммерческих программных пакетов, основанных на моделях дерева отказов, используются инженерами по надежности в повседневной практике в различных отраслях промышленности (атомная, нефтяная, аэронавтика, автомобилестроение и т.д.). Они могут быть использованы для расчета PFD_{avg}, но аналитик должен быть очень осторожным, потому что некоторые из них реализуют вычисление PFD_{avg} некорректно. Основной ошибкой является неправильное вычисление сочетания $PFD_{avg,i}$ отдельных компонентов (как правило, получаемых просто как $\frac{\lambda_i \tau}{2}$) для получения предполагаемого результата для PFD_{avg} всей системы. Как было показано выше, результат оказывается неверным и неконсервативным.

Во всяком случае, программные пакеты, основанные на методе дерева отказов, могут быть использованы для вычисления мгновенной неготовности системы $U_{Sf}(t)$ исходя из мгновенной неготовности компонентов $U_i(t)$. После этого может быть вычислено среднее значение $USf(t)$ за определенный период времени для нахождения PFD_{avg}. В зависимости от используемого программного обеспечения это может быть сделано самим программным пакетом или используя дополнительные вычисления.

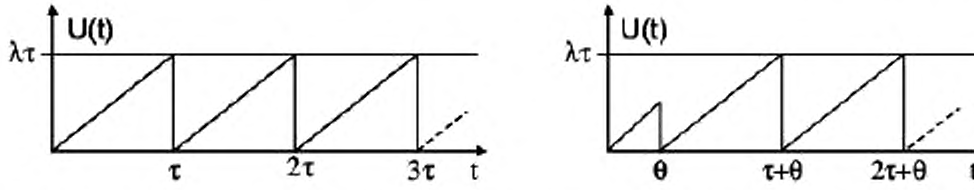


Рисунок В.19 – Мгновенная неготовность $U(t)$ отдельного периодически проверяемого элемента.

Ранее описанный идеальный случай показан слева на рисунке В.19:

$$U_i(t) = \lambda\zeta \quad \text{и} \quad \zeta = t \ \text{по модулю} \ \tau .$$

Эта так называемая «зубчатая» кривая увеличивается линейно от 0 до $\lambda\tau$ и начинается снова с 0 после испытания или ремонта (которые считаются мгновенными, т.к. в это время УО не работает).

Когда в структурах с резервированием используются несколько компонентов, испытания могут иметь график, как показано справа на рисунке В.19, где первый интервал проверки отличается от других. Это не влияет на PFD_{avg} или на максимальные значения, которые равны $\frac{\lambda\tau}{2}$ и $\lambda\tau$ в этих случаях.

Конечно, в неидеальном случае кривые могут быть более сложными, чем показаны на рисунке. В В.5.2 будут даны руководящие указания по проектированию более точных зубчатых кривых, но для целей данного пункта вид кривых, представленных на рисунке В.19, вполне приемлем.

На рисунке В.20 проиллюстрировано применение данного подхода к небольшому дереву отказов, представленному на рисунке В.18 (на рисунке В.20 *DU* означает необнаруженные опасности, а *CCF* – отказы по общей причине). Мы упоминали, что система имеет два дублирующих компонента (*E* и *F*) и что *D* – общая причина отказа этих компонент. Для вычисления использовались следующие значения:

Коэффициент β был выбран так, чтобы быть уверенным, что *CCF* не привалирует в результирующем значении PFD_{avg}, и чтобы получить лучшее понимание, как в данном методе вычисляется PFD_{avg}.

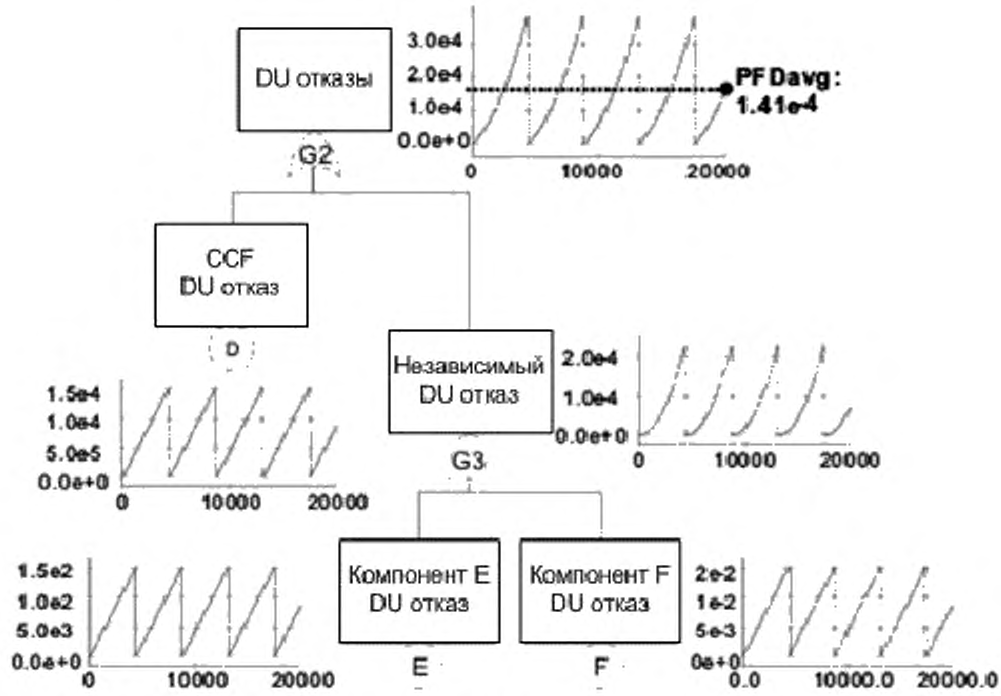


Рисунок В.20 – Принцип вычисления PFD_{avg} при помощи дерева отказов

Нетрудно понять, что вид зубчатой кривой на входах *D*, *E* и *F* аналогичен левой кривой на рисунке В.19. *CCF* (*D*) проверяется каждый раз, когда проверяется *E* или *F*. *E* и *F* проверяются в одно и то же время каждые 6 месяцев, *CCF* (*D*) также проверяется каждые 6 месяцев.

Используя один из алгоритмов, разработанных для вычисления дерева отказов, довольно просто сформировать зубчатую кривую на выходах каждого логического блока. PFD_{avg} вычисляется путем усреднения результатов, полученных для события верхнего уровня. Это может быть выполнено, если использовать как программное обеспечение метода, так и расчет вручную. Полученное значение для $PFD_{avg} = 1{,}4 \times 10^{-4}$ в соответствии с настоящим стандартом соответствует уровню УПБ 3 для режима работы с низкой интенсивностью запросов.

Как показано на рисунке В.20, графики между проверками сглаживаются. Поэтому вычислить среднее значение несложно при условии, что определен и учтен момент проверки.

Интересно отметить, что если реализовано резервирование, то зубчатые кривые событий верхнего уровня между проверками становятся нелинейными (т.е.

интенсивность отказа всей системы больше не является постоянной величиной).

Также интересна оценка влияния на PFD_{avg} сдвига по времени испытаний дублирующих компонент вместо проведения их в одно и то же время. Это показано на рисунке В.21, где проверки компоненты *F* сдвинуты по времени от проверок компоненты *E* на три месяца.

Это приводит к ряду важных последствий:

- *CCF* теперь проверяется каждые три месяца (т.е. каждый раз, когда проверяют *E* и каждый раз, когда проверяют *F*). Частота контрольных испытаний в два раза выше, чем в предыдущем случае.

- Зубчатый график события верхнего уровня также имеет частоту контрольных проверок в два раза выше, чем раньше.

- Зубчатый график имеет меньшие отклонения относительно среднего значения по сравнению с предыдущим случаем.

- *PFDavg* снизилось до значения 8,3 × 10^{-5}: с этой новой политикой проверки система достигла УПБ 4.

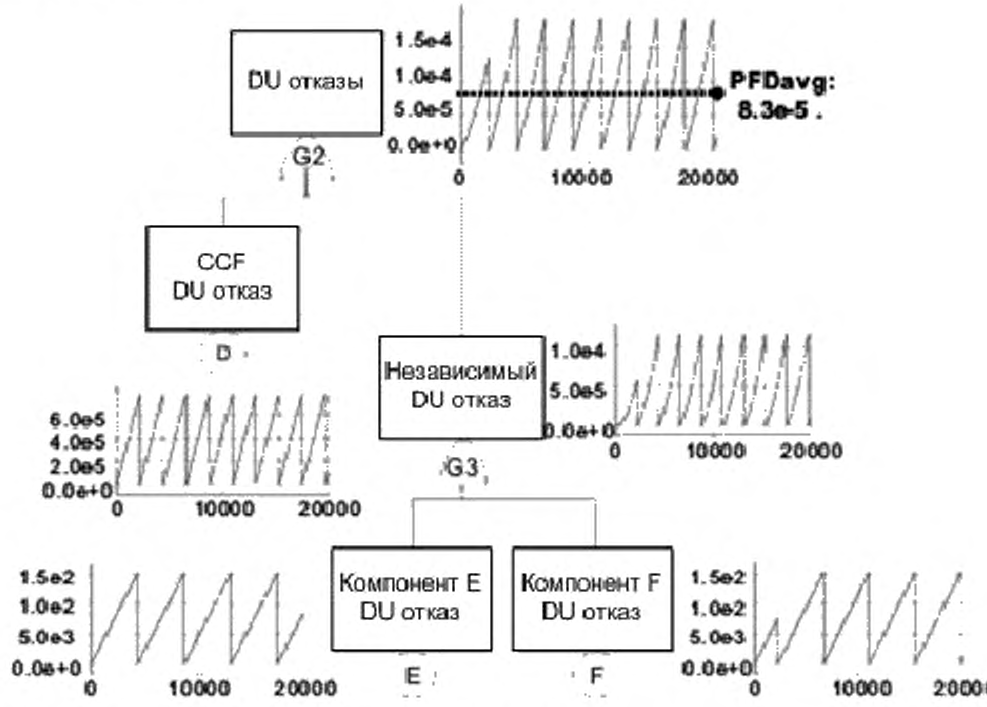


Рисунок В.21 – Влияние смещения проверок

Если проверки смещены относительно друг друга и реализованы корректные

процедуры, то это увеличит вероятность обнаружения *CCF* и является эффективным методом уменьшения *CCF* для систем, работающих в режиме низкой интенсивности запросов. Это позволило улучшить значение УПБ с УПБ 3 до УПБ 4 (для отказов аппаратных средств и при условии, что выполнены другие требования МЭК 61508).

Рисунок В.22 представляет зубчатую кривую, полученную при последовательном добавлении к системе, смоделированной на рисунке В.20, элемента *G* (при отсутствии проверок) и элемента *H* (с контрольными проверками каждые два года).

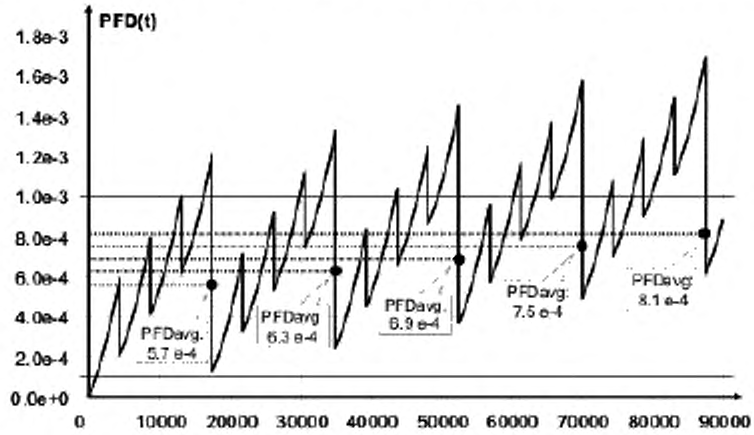


Рисунок В.22 – Пример комплексного шаблона проверки

Влияние никогда не проверяемого элемента *G* двоякое: *PFD(t)* никогда не примет нулевое значение, если проверки проводятся каждые два года и значение PFD_{avg} непрерывно возрастает (черные точки соответствуют PFD_{avg} в течение периода, ограниченного соответствующей пунктирной линией).

Даже если простейшие зубчатые кривые (как, например, показаны на рисунке В.19) довольно просты, результаты для события верхнего уровня могут быть довольно сложными, но это не затрудняет применение метода.

Цель настоящего пункта заключается только в демонстрации принципа расчета с использованием логических моделей. В.5.2, относящийся к марковскому подходу, содержит ряд руководящих указаний по созданию более сложных входных зубчатых кривых для элементарных компонент.

Можно сделать вывод о том, что если отдельные компоненты являются относительно независимыми, то нет проблем при вычислении PFD_{avg} для Э/Э/ПЭ системы, связанной с безопасностью, с помощью классических логических методов. С теоретической точки зрения это не так просто, и аналитик, проводящий исследование, должен иметь глубокие знания о вероятностных методах, чтобы выявить и отклонить

иногда встречающиеся некорректные значения PFD_{avg}. При условии выполнения этих мер предосторожности может быть использован любой программный пакет для расчета деревьев отказов.

Для расчетов *PFH* также могут быть использованы логические методы, но их теоретическое обоснование выходит за рамки данного приложения.

B.5 Подходы, основанные на моделях состояний/переходов

B.5.1 Основные положения

Логические модели в основном не зависят от времени и введение понятия времени возможно только в некоторых особых случаях. Они довольно искуственны и, чтобы избежать ошибок, требуют хорошего знания вероятностных методов. Поэтому в таких случаях могут быть использованы другие вероятностные модели, динамические по природе. В области надежности они основаны на следующем фундаментальном подходе, состоящем их двух этапов:

- определение всех состояний системы на этапе изучения;
- анализ переходов системы из состояния в состояние в соответствии с происходящими событиями и на протяжении их существования.

Именно поэтому они находятся в категории моделей состояний-переходов.

Основной подход состоит в построении для изучаемой системы некоторой модели поведения автомата с возникающими событиями (отказами, ремонтами, испытаниями и т.д.). В настоящем стандарте считается, что Э/Э/ПЭ системы, связанные с безопасностью, имеют только дискретные состояния. Данные модели являются динамическими по своей природе и могут быт реализованы различными способами: графическим представлением, специальным формальным языком или универсальным языком программирования. В данном приложении представлены два из них, которые существенно различаются, но дополняют друг друга:

- модель Маркова, которая разработана в самом начале прошлого века. Она хорошо изучена и обрабатывается аналитически;
- сеть Петри, которая была разработана в 60-х годах. Она менее известна (но все больше и больше используется из-за ее гибкости) и применяется совместно с моделированием методом Монте-Карло.

Оба способа основаны на графическом представлении, что очень удобно пользователям. Другие методы основываются на моделях, лежащих в основе формальных зыков, что будет кратко рассмотрено в конце данного раздела.

B.5.2 Подход Маркова

B.5.2.1 Принцип моделирования

Марковский подход является самым известным из всех динамических подходов в области надежности. Марковские процессы разделяются на гомогенные (или однородные процессы, в которых все интенсивности переходов являются постоянными величинами) и прочие (полумарковские процессы). Т.к. будущее гомогенного процесса Маркова не зависит от его прошлого, то выполняемые аналитические вычисления являются относительно простыми. Для более сложного, полмарковского процесса может быть использован метод моделирования Монте-Карло. Настоящий стандарт рассматривает только гомогенные процессы и для упрощения термин «марковские процессы» используется именно в этом смысле (см. МЭК 61508-7, п. C.6.4 и [5]).

Основная базовая формула для марковских процессов

$$P_i(t+dt) = \sum_{k \neq i} P_k(t)\lambda_{ki}dt + P_i(t)\left(1 - \sum_{k \neq i} \lambda_{ik}dt\right),$$

где λ_{ki} – интенсивность перехода (т.е. частота отказов или ремонтов) из состояния *i* в состояние *k*. Понятно, что вероятность нахождения в состоянии *i* в момент времени *t+dt* является вероятностью перехода к состоянию *k* (если другим состоянием является *k*) или вероятностью пребывания в состоянии *i* (если система уже находится в этом состоянии) на протяжении времени с *t* по *t+dt*.

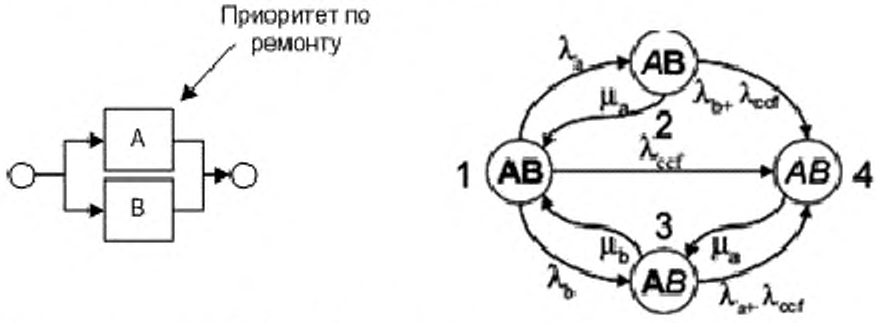


Рисунок B.23 – Граф марковской модели, описывающей поведение системы из двух компонент

Существует тесная связь между формулой, указанной выше, и графическим представлением на рисунке B.23, которое моделирует систему, состоящую из двух компонент с одной командой ремонта (компонент A имеет больший приоритет на ремонт) и общей причиной отказа. На данном рисунке **A** обозначает, что компонент A – в рабочем состоянии, ***A*** – в состоянии отказа. Так как необходимо учитывать время обнаружения, μ_a и μ_b на рисунке B.23 являются частотами восстановления компонентов

(т.е. μ_a =1/*MTTR*$_a$ и μ_b =1/*MTTR*$_b$).

Например, вероятность нахождения в состоянии 4 просто вычисляется по следующей формуле

$$P_4(t+dt) = \left[P_1(t)\lambda_{ccf} + P_2(t)\left(\lambda_b + \lambda_{ccf}\right) + P_3(t)\left(\lambda_a + \lambda_{ccf}\right)\right]dt + P_4(t)(1-\mu_a dt).$$

Что приводит к дифференциальному уравнению в векторной форме

$d\vec{P}(t)/_{dt} = [M]\vec{P}(t)$, которое условно разрешимо:

$\vec{P}(t) = e^{t[M]}\vec{P}(0)$,

где *M* – матрица Маркова, содержащая частоты переходов, а $\overrightarrow{P(0)}$ – вектор начальных условий (обычно вектор-столбец с 1 для начального состояния и 0 для остальных).

Даже если экспонента матрицы имеет не точно такие же свойства, как обычная экспонента, то можно записать:

$$\vec{P}(t) = e^{(t-t1)[M]}e^{t1[M]}\vec{P}(0) = e^{(t-t1)[M]}\vec{P}(t1).$$

Это показывает базовое свойство марковских процессов: знание вероятностей состояний в заданный момент времени *t1* есть сумма знаний всех предыдущих и этого достаточно для вычисления поведения системы после момента времени *t1*. Это очень полезно для вычисления *PFD*.

Для решения указанных уравнений уже давно были разработаны эффективные алгоритмы и реализованы в программных пакетах. Поэтому при использовании данного подхода аналитик может только строить модели и не вникать в лежащую в основе математику, хотя в любом случае он должен понимать, по крайней мере то, что представлено в данном приложении.

Рисунок В.24 показывает принцип расчета *PFD*:

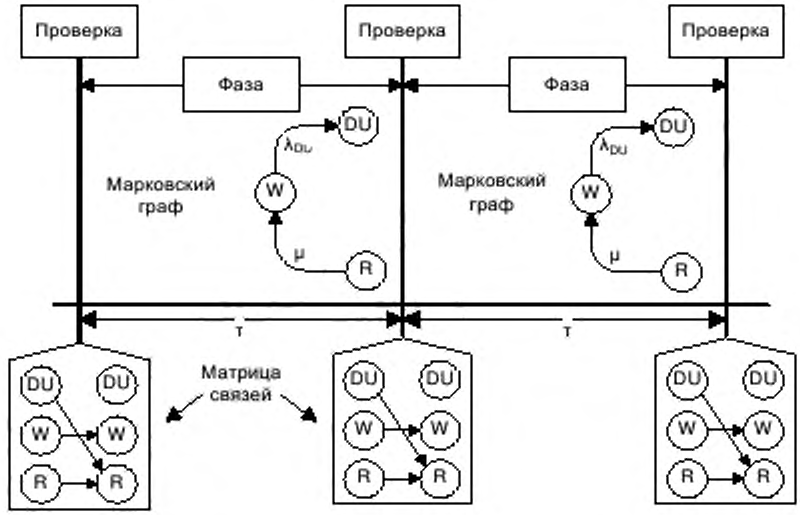


Рисунок В.24 – Принцип марковского многофазного моделирования

Расчеты *PFD* относятся к Э/Э/ПЭ системам, связанным с безопасностью, работающим в режиме работы с низкой интенсивностью запросов и с периодическими (контрольными) проверками. Для подобных систем ремонты начинаются только после проведения проверок. Моменты выполнения проверок являются особыми точками на временной оси, но многофазный подход Маркова может быть использован для решения этой проблемы.

Например, в простой системе производится периодическая проверка одного компонента, имеющего три состояния, как показано на рисунке В.24: рабочее, необнаруженный опасный отказ (*DU*) и ремонт (*R*).

Его поведение между испытаниями моделируется марковским процессом, показанным сверху на рисунке В.24: он может отказать (*W* -> *DU*) или быть в ремонте (*R* ->*W*). Так как ремонт не может начаться во время интервала проверки, то нет и перехода от *DU* к *R*. Ввиду того, что диагностика отказа производится после перехода в состояние *R*, μ является частотой ремонта компонента (т.е. $\mu = 1/MRT$) на рисунке В.24.

Когда испытание уже проведено (см. матрицу связей на рисунке В.24), начинается ремонт, если произошел отказ (*DU*->*R*), или компонент продолжает работать, если он находится в нормально функционирующем состоянии (*W*->*W*) и в совсем уж гипотетической ситуации, когда ремонт, который начался еще после предыдущей проверки, еще не завершился и продолжается (*R*->*R*). Матрица связей [*L*] может быть

использована для вычисления начальных условий в начале состояния *i*+1 из вероятностей состояний в конце состояния i. В результате получаем следующее уравнение:

$$\begin{bmatrix} P_{DU}(0) \\ P_W(0) \\ P_R(0) \end{bmatrix}_{i+1} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{DU}(\tau) \\ P_W(\tau) \\ P_R(\tau) \end{bmatrix} \equiv \overrightarrow{P_{i+1}}(0) = [L]\overrightarrow{P_i}(\tau)$$

Замена $\overrightarrow{P_i}(\tau)$ его значением приводит уравнение к реккурентному виду, что позволяет вычислить начальные условия в начале каждого проверочного интервала:

$$\overrightarrow{P_{i+1}}(0) = [L]e^{\tau[M]}\overrightarrow{P_i}(0).$$

Данное выражение может быть использовано для вычисления вероятностей в любой момент времени $t = i\tau + \zeta$. Например, во время тестового интервала *i* получается следующее выражение

$$\vec{P}(t) = \vec{P}_i(\zeta) = e^{\zeta[M]}\vec{P}_i(0), \quad (i-1)\tau \le t < i\tau, \quad \zeta = t \bmod \tau .$$

Получить значение мгновенной недоступности можно путем простого суммирования вероятностей состояний, когда система недоступна. Для выражения мгновенной недоступности полезен линейный вектор (q_k)

$$U(t) = \sum_{k=1}^{n} q_k P_k(t) ,$$

где q_k=1 означает, что система недоступна в состоянии *k*, иначе q_k=0.

Для простой модели получается выражение $PFD(t) = U(t) = P_{DU}(t) + P_R(t)$ и зубчатая кривая выглядит, как на рисунке В.25.

PFD_{avg} вычисляется описанным ранее способом через *MDT*, что легко получить из среднего учтенного времени (*MCT*), проведенного системой в этих состояниях

$$\overrightarrow{MCT}(T) = \int_0^T \vec{P}(t)dt .$$

Как и для $\vec{P}(t)$, существуют эффективные алгоритмы для вычисления этого интеграла на отрезке [0, *T*] и окончательно получаем: $PFD_{avg}(T) = \frac{1}{T}\sum_{k=1}^{n} q_k MCT_k(T)$.

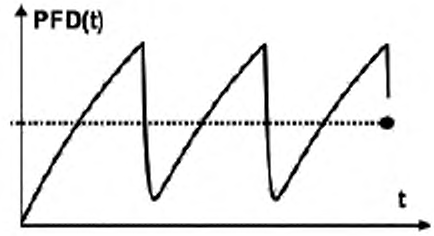


Рисунок В.25 – Зубчатая кривая, полученная с помощью многофазного марковского подхода

Применяя данную формулу к модели, показанной на рисунке В.24, получаем:

$$PDF_{avg}(T) = \frac{1}{T}[MCT_{DU}(T) + MCT_{R}(T)]$$

Из формулы можно убрать первое слагаемое, если УО был выключен в момент ремонта.

Черная точка на рисунке В.25 – это PFD_{avg} зубчатой кривой для всего периода вычислений.

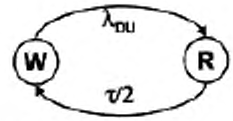


Рисунок В.26 – Приближенная марковская модель

Необходимо отметить, что указанные выше вычисления обычно производятся с использованием приближенной марковской модели, показанной на рисунке В.26, где состояния *DU* и *R* связаны и где $\tau/2$ (т.е. среднее время до обнаружения отказа) используется как эквивалент времени восстановления. Это верно, если только уравнение Маркова было решено ранее другим методом в целях нахождения данного равенства. Приближение применимо, только если время ремонта незначительно. Этот метод может встретить большие затруднения для больших сложных систем.

Простая модель, показанная на рисунке В.24, может быть легко усовершенствована для более реальных компонентов. На рисунке В.27 матрица связи моделирует компонент, который как с вероятностью γ может оказаться в состоянии отказа при запросе (то есть реальный отказ при запросе), так и с вероятностью σ может оказаться, что отказ не был обнаружен при проверке (вследствие ошибки человека).

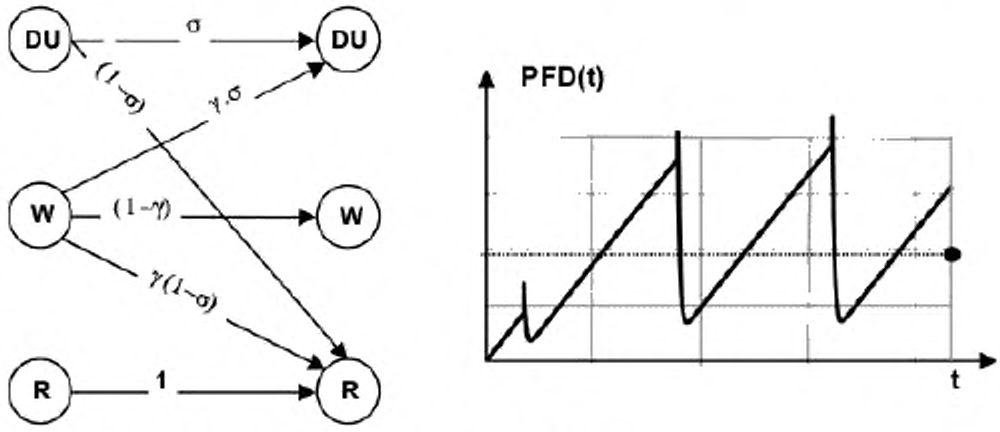


Рисунок B.27 – Влияние на отказы самого запроса

Вид зубчатой кривой изменился, и скачки, наблюдаемые для каждой проверки, соответствуют вероятности отказа γ. И опять черная точка представляет собой PFD_{avg}.

Когда (резервный) компонент отключен для проверки, он становится недоступным во время всего проведения испытания и это влияет на его PFD_{avg}. Таким образом, должна быть учтена продолжительность испытания π и введена дополнительная фаза между проверочными интервалами. Это показано на рисунке B.28, где состояния *R* и *W* смоделированы в данной фазе только ради полноты картины.

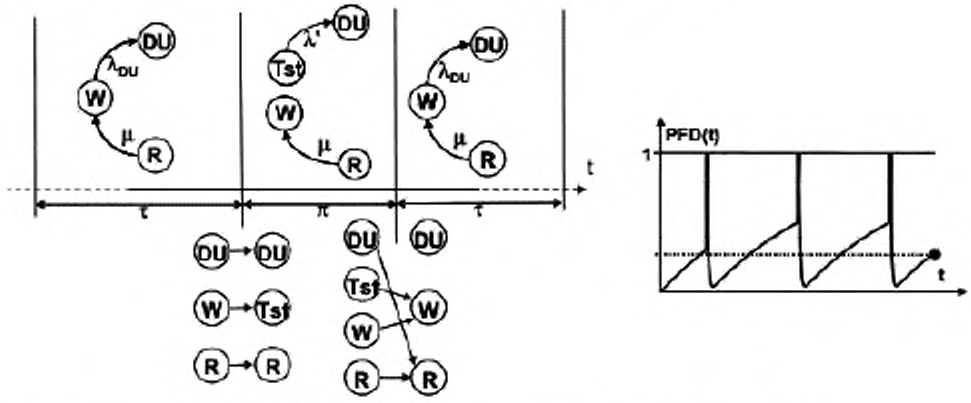


Рисунок B.28 – Моделирование влияния длительности проверки

В данной марковской модели система недоступна в состояниях *R*, *DU* и *Tst*. Это несколько сложнее, чем раньше, но принцип расчетов остался точно таким же. Поведение зубчатой кривой показано на рисунке B.28 справа – система недоступна во время проверок и это может быть основным вкладом в PFD_{avg}.

На предыдущем марковском графе рассматривались только опасные необнаруженные отказы, но опасные обнаруженные отказы тоже могут быть представлены. Отличие в том, что ремонт начинается точно в тот момент, как показано на рисунке В.29. Таким образом, μ_{DD} – это интенсивность восстановления компонента ($\mu_{DD} = 1/MTTR$), а μ_{DU} – интенсивность ремонта ($\mu_{DU} = 1/MRT$).

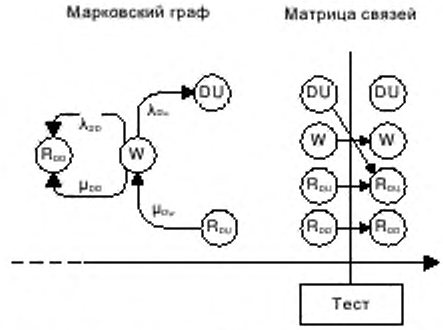


Рисунок В.29 – Многофазная модель Маркова с отказами *DD* и *DU*

В случае необходимости должны быть отражены и безопасные отказы, но в данном случае для простоты это не делается.

Основная проблема с марковскими графами в том, что количество состояний растет экспоненциально при увеличении количества компонентов изучаемой системы. Так что построение марковских графов и проведение указанных выше расчетов вручную без серьезного приближения очень быстро становится невыполнимым.

Использование эффективных программных пакетов для марковских моделей помогает справиться со сложностями вычислений. Существует огромное количество доступных пакетов, даже если они не обязательно непосредственно применимы для вычислений PFD_{avg}: большинство из них предназначены для вычисления мгновенной недоступности, но только некоторые из них вычисляют среднее накопленное время нахождения системы в конкретных состояниях, и только единицы позволяют прозводить многофазное моделирование. В любом случае их не так сложно адаптировать для вычисления PFD_{avg}.

Что касается самого моделирования, если зависимости между компонентами слабые, марковский и логический подходы могут быть объединены:

- марковская модель может быть использована для установления мгновенной недоступности для каждого из компонентов;

- деревья отказов или блок-схемы надежности используются для объединения

отдельных недоступностей для вычисления мгновенной недоступности *PFD(t)* всей системы;

- PFD_{avg} получается путем усреднения *PFD(t)*.

Такой подход описан в В.4 и зубчатые кривые, подобные тем, что показаны на рисунках В.25, В.27 и В.28, могут быть использованы как входные данные для деревьев отказов.

Если зависимостями между компонентами нельзя пренебречь, то можно воспользоваться каким-либо инструментом для автоматического построения марковского графа. Они основываются на моделях более высокого уровня, чем марковские (например, сети Петри или формальный язык). Из-за комбинаторного взрыва числа состояний они все равно могут столкнуться с трудностями.

Объединенный подход очень эффективен для моделирования сложных систем.

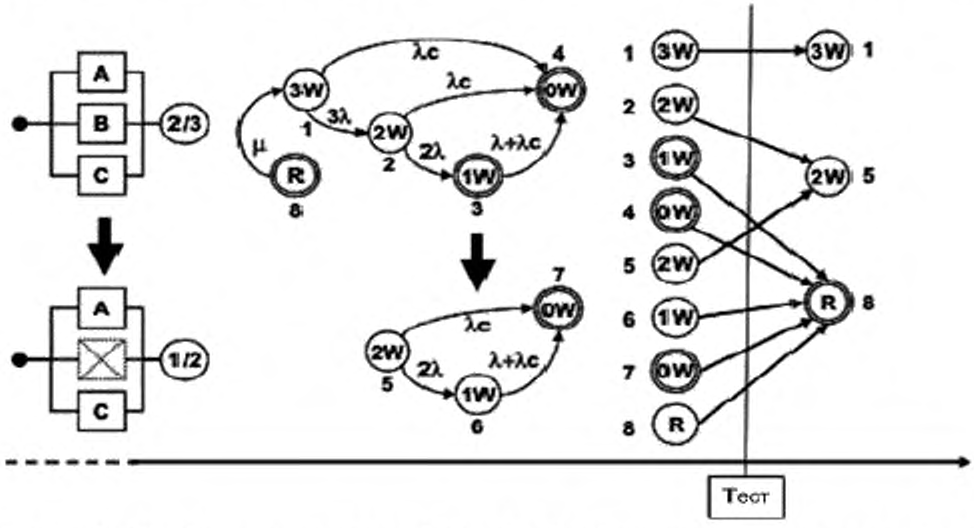


Рисунок В.30 – Изменение логики (с 2оо3 на 1о2) вместо ремонта первого отказа

Система, смоделированная на рисунке В.30, состоит из трех компонентов, проверяемых в одно и то же время и работающих по схеме 2оо3. Когда отказ обнаружен, логика меняется с 2оо3 на 1оо2, т.к. логика 1оо2 лучше, чем 2оо3 с точки зрения безопасности (но хуже с точки зрения ложного отказа). И только в случае обнаружения второго отказа должен произойти ремонт, который включает в себя замену всех трех элементов на новые. Это вносит системные ограничения, поэтому невозможно построить поведение всей системы простым объединением поведения независимых компонентов.

B.5.2.2 Принцип расчета *PFH*

Такой же тип мультифазного марковского моделирования может быть использован для расчета *PFH* для *DU*-отказов, обнаруженных контрольными проверками. В целях упрощения будет показан только принцип вычисления *PFH* для *DD* отказов, для которых нужны только обычные (однофазные) модели Маркова. Конечно, для Э/Э/ПЭ систем, связанных с безопасностью, работающих в режиме с непрерывным запросом и имеющих обнаруженные периодическими контрольными проверками *DU* отказы, должен быть использован многофазный марковский подход. Это не меняет принцип, рассматриваемый ниже.

Рисунок B.31 показывает два марковских графа, моделирующих одну и ту же систему, сделанную из двух дублирующих компонентов с общей причиной отказа. С левой стороны компоненты (A и B) могут быть отремонтированы. С правой стороны – нет.

На обоих графах состояние 4 (*AB*) является поглощающим. Система остается отказавшей после отказа всей системы и *P(t) = P1(t)+P2(t)+P3(t)* является вероятностью того, что не произошло отказа на промежутке [0; t]. Тогда *R(t) = P(t)* является надежностью системы, а *F(t) = 1-R(t)=P4(t)* является ее ненадежностью.

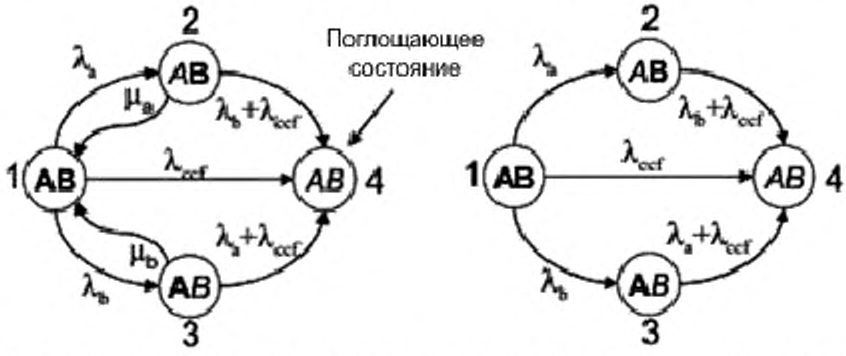


Рисунок B.31 – Марковский граф «надежности» с поглощающим состоянием

Как обсуждалось в B.2.3, возможно использование модели надежности для обработки ситуации, когда отказ Э/Э/ПЭ системы, связанной с безопасностью, сразу приводит к опасной ситуации. Опять же, μ_a и μ_b являются интенсивностями восстановления компонентов (т.е. $\mu_a = 1/MTTRa$ и $\mu_b = 1/MTTRb$).

Такой марковский граф надежности позволяет получить *PFH* непосредственно *PFH=F(T)/T*. Например, из рисунка B.31 можно непосредственно получить *PFH(T) = P4(T)/T* (при условии, что *P4(T)*<<1).

Так же такой марковский граф надежности позволяет вычислять *MTTF* системы по следующей формуле

$$MTTF = \lim_{t\to\infty} \sum_{k=1}^{n} a_k MCT_k(t).$$

В данной формуле $MCT_k(t)$ является средним накопленным временем нахождения в состоянии *k*, a_k=1, если *k* является нормальным рабочим состоянием, a_k=0 во всех других случаях.

Верхние границы можно получить следующим образом:

$$PFH \approx \frac{1}{MTTF}.$$

Эффективные алгоритмы расчета известны и почти все программные пакеты для марковских моделей могут быть использованы для расчетов *F(T)* и *MTTF*.

Указанные выше ожидания для *PFH* верны для всех случаев, даже если интенсивность отказов всей системы непостоянна (как для графа, расположенного справа на рисунке B.31). Единственное ограничение состоит в том, что нужно использовать марковский граф надежности с одним (или несколькими) поглощающим(и) состоянием(ями). Конечно, это верно и при использовании многофазных моделей.

Когда все состояния являются полностью и быстро восстановимыми, общая интенсивность отказов системы быстро стремится к асимптотическому значению Λ_{as} = 1/*MTTF*. В таких графах, за исключением исходных и поглощающих состояний, все остальные являются квазимгновенными (так как значения *MTTR* для компонентов являются существенно меньше их *MTTF*). Это позволяет непосредственно вычислять постоянные интенсивности отказов всей системы для каждого сценария, начиная от исходного и заканчивая поглощающим состоянием. Марковский граф слева на рисунке B.31 моделирует такую полностью и быстро восстанавливаемую систему. Так что:

$1\to4$: $\Lambda_{14}=\lambda_{ccf}$

$1\to2\to4$: $\Lambda_{124}=\lambda_a.(\lambda_b+\lambda_{ccf})/[(\lambda_b+\lambda_{ccf})+\mu_a]\approx\lambda_a.(\lambda_b+\lambda_{ccf})/\mu_a$

$1\to3\to4$: $\Lambda_{134}=\lambda_b.(\lambda_a+\lambda_{ccf})/[(\lambda_a+\lambda_{ccf})+\mu_b]\approx\lambda_b.(\lambda_a+\lambda_{ccf})/\mu_b$.

В формуле для сценария $1\to3\to4$ λ_b – вероятность перехода из исходного состояния, а $(\lambda_a + \lambda_{ccf})/\mu_b$ – вероятность перехода в состояние 4 предпочтительнее возврата в состояние 1, если система оказалась в состоянии 3.

Наконец: $\Lambda_{as} = \Lambda_{12} + \Lambda_{124} + \Lambda_{134} = \frac{1}{MTTF}$.

Это можно легко обобщить для сложных марковских графов, но оно верно лишь

для полностью и быстро восстановимых систем, т.е. *DD* отказов.

Марковский граф справа на рисунке В.31 не является полностью и быстро восстановимым. Так что использование приведенных выше вычислений даст неправильный результат.

Если Э/Э/ПЭ система, связанная с безопасностью, работающая в режиме с непрерывным запросом, используется вместе с другими слоями безопасности, то должна быть рассмотрена ее готовность. Это показано на обоих графах на рисунке В.32: там нет поглощающего состояния и система восстанавливается после полного отказа. *P(t) = P1(t) + P2(t) + P3(t)* – вероятность того, что система работает в момент времени *t*. Тогда *A(t) = P(t)* является готовностью, а *U(t) = 1 - A(t) = P4(t)* – неготовностью.

Данный случай существенно отличается от примера, приведенного на рисунке В.31, поэтому *R(t)* и *A(t)* должны использоваться корректно, так же, как *U(T)* и *F(T)*, если необходимо получить корректные результаты.

В случае *DD* отказов простейшим путем решения такой проблемы является вычисление верхней границы *PFH* через *MDT* и *MUT*, как показано в В.2.3.

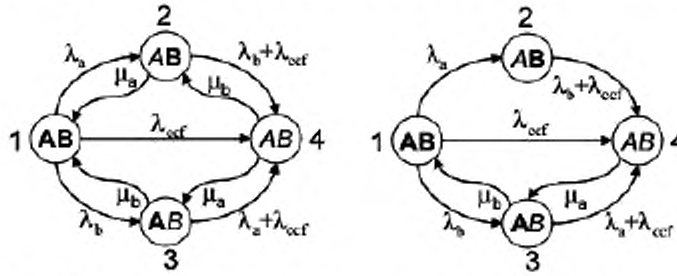


Рисунок В.32 – Марковский граф «готовности» без поглощающих состояний

Интересное свойство марковских графов готовности состоит в том, что они достигают асимптотического равновесия, когда вероятность перехода в данное состояние равна вероятности перехода из него. Заметим:

- $P_{i,as} = \lim_{t \to \infty} P_i(t)$ – асимптотическое значение $P_i(t)$;
- $\lambda_i = \sum_{j \neq i} \lambda_{ij}$ – вероятность переходов из состояния i в любое другое.

Каждый раз, когда система переходит в состояние i, среднее время нахождения в этом состоянии равно $Mst_i = \frac{1}{\lambda_i}$.

Это позволяет вычислить $MUT = \sum_i (1 - q_i) P_{i,as} Mst_i$ и $MDT = \sum_i q_i P_{i,as} Mst_i$, где q_i=0,

если i – рабочее состояние, и q_i=1 в противном случае.

В результате получаем следующее выражение:

$$PFH = 1/(MUT + MDT) = 1/\sum_i P_{i,as} Mst_i = 1/\sum_i \frac{P_{i,as}}{\lambda_i}.$$

Необходимо отметить, что количество отказов, произошедших в период [0, T], равно: $n = \frac{T}{\sum_i \frac{P_{i,as}}{\lambda_i}}$.

В большинстве случаев программные пакеты могут найти асимптотические вероятности, так как нет каких-то особых сложностей с выполнением подобных вычислений.

Если период рассмотрения слишком маленький для сходимости марковского процесса, то *PFH* может быть получен как $w(t) = \sum_{i \neq f} \lambda_{if} P_i(t)$. В результате получаем:

$$PFH(T) = \sum_{i \neq f} \lambda_{if} \frac{\int_0^T P_i(t)dt}{T} = \frac{\sum_{i \neq f} \lambda_{if} MCT_i(T)}{T}.$$

Нет никакой сложности в том, чтобы произвести подобные вычисления в специальном программном пакете, подставив накопленное время для каждого из состояний.

В случае полностью и быстро восстанавливаемых систем (*DD* отказы), функция интенсивности отказов Веселя (Vesely) $\Lambda_v(t)$ очень быстро сходится к асимптотическому значению Λas, которое является хорошим приближением постоянной интенсивности отказа всей системы. Таким образом *PFH* в этом случае может быть получена так же, как и для безотказности.

Случай *DU* отказов является наиболее сложным ввиду многофазного моделирования. Указанная выше формула может быть обобщена:

$$PFH(T) = \frac{\sum_{\varphi=1}^{n} \sum_{i \neq f} \lambda_{if} MCT_i(T_\varphi)}{\sum_{\varphi=1}^{n} T_\varphi}.$$

В данной формуле T_φ является длительностью фазы φ.

Многофазные марковские процессы обычно достигают равновесия, когда вероятность перехода из заданного состояния равна вероятности перехода в него.

Асимптотические значения не имеют ничего общего с теми, которые описаны выше, но они могут быть использованы в приведенной выше формуле.

Необходимо отметить, что марковский подход предоставляет множество возможностей для вычисления *PFH* Э/Э/ПЭ системы, связанной с безопасностью, работающей в режиме с непрерывным запросом. Однако для корректного применения марковского подхода необходимо хорошее понимание лежащего в его основе математического аппарата.

В.5.3 Сети Петри и метод моделирования Монте-Карло

В.5.3.1 Принцип моделирования

Эффективным способом моделирования динамических систем является создание конечного автомата, поведение которого настолько близко к поведению изучаемой Э/Э/ПЭ системы, связанной с безопасностью, насколько это возможно. Сети Петри (см. МЭК 61508-7, п. В.2.3.3 и п. В.6.6.10), как было доказано, являются очень эффективным средством для этой цели по следующим причинам:

- их легко обрабатывать графически;
- размер моделей растет линейно относительно количества моделируемых компонентов;
- они очень гибкие и позволяют моделировать большинство ограничений;
- они идеально подходят для моделирования с использованием метода Монте-Карло (см МЭК 61508-7, п. В.6.6.8).

Разработанные в 1960-х годах для формального доказательства в теории автоматов, они были активно применены инженерами по надежности для решения двух задач: автоматизации построения больших графов Маркова и в 80-х – для моделирования с использованием метода Монте-Карло.

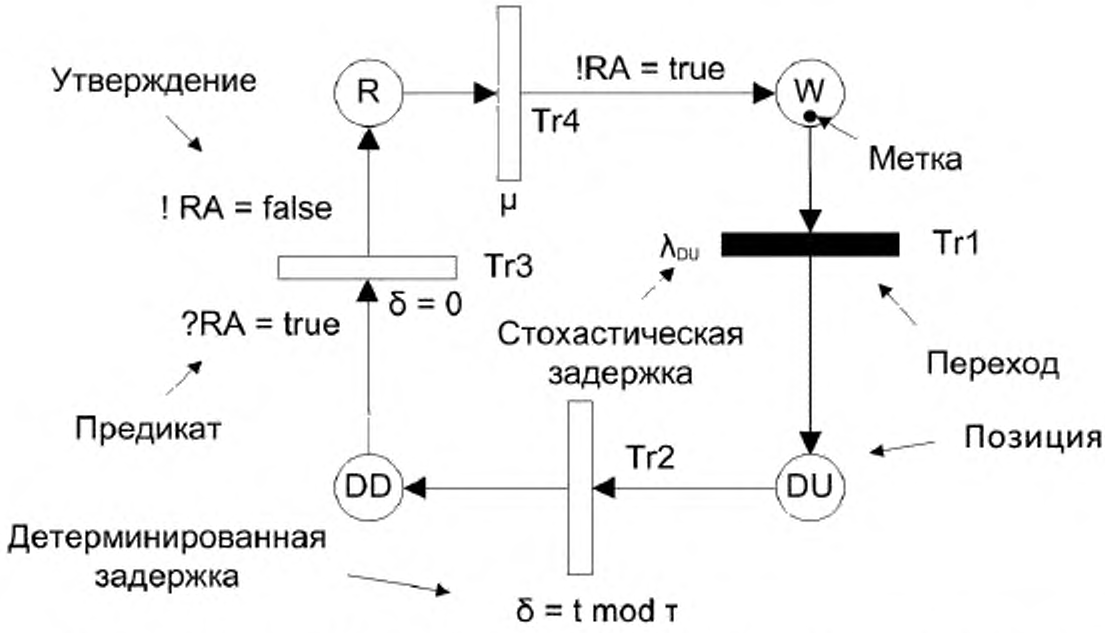


Рисунок В.33 – Сеть Петри для моделирования одного периодически проверяемого компонента

Типичная подсеть Петри для простого периодически проверяемого компонента состоит из трех частей:

1) Статическая часть (т.е. рисунок):
   a) позиции (круги) соответствуют возможным состояниям;
   b) переходы (прямоугольники) соответствуют возможным событиям;
   c) стрелки вверх (от позиций к переходам) разрешают переходы;
   d) стрелки вниз (от переходов к позициям) показывают, что происходит, когда запускается переход.
2) Часть планирования:
   a) стохастические задержки являются случайными задержками, произошедшими до события;
   b) детерменированные задержки – это известные задержки, произошедшие до события.
3) Динамическая часть:
   a) метки (маленькие черные точки), которые двигаются, когда происходит событие, для отображения того, какое из возможных состояний достигнуто;
   b) предикаты (любая формула, которая может быть истинной или ложной),

разрешающие переходы;

c) утверждения (любые выражения), обновляющие какие-либо переменные, когда переход запускается.

Кроме того, существует ряд правил разрешения и запуска перехода:

4) Разрешение перехода (т.е. условия для соответствующего события, которое может произойти):

a) все входные позиции имеют как минимум одну метку;

b) все предикаты должны быть истинными.

5) Запуск перехода (т.е. что происходит, когда соответствующее событие реализуется):

a) одна метка удаляется из входной позиции;

b) одна метка добавляется в выходную позицию;

c) утверждения обновляются.

Большинство понятий, связанные с сетями Петри, введены выше, остальные будут вводиться по мере необходимости.

B.5.3.2 Принцип моделирования Монте-Карло

Моделирование Монте-Карло представляет собой анимацию моделей поведения с помощью случайных чисел, чтобы определить, как часто система остается в состояниях, управляемых или случайными или детерминированнными задержками (см. также МЭК 61508-7, п. B.6.6.8).

Это можно объяснить с помощью сети Петри, представленной на рисунке B.33:

- Вначале метка находится в позиции *W* и компонент находится в нормально работающем состоянии.

- Только одно событие может появиться в данном состоянии – опасный необнаруженный отказ (переход Tr1 разрешен и закрашен черным).

- Время, проведенное в данном состоянии, является случайной величиной и зависит от экспоненциального распределения параметра λ_{DU}. Метод Монте-Карло состоит в использовании случайных чисел (см. ниже) для вычисления задержки *d1* перед отказом, который должен произойти (т.е. должен быть запущен переход Tr1).

- Когда время *d1* вышло, запускается переход Tr1 и метка перемещается в позицию *DU* (точнее, одна метка удаляется из позиции *W* и одна метка добавляется в позицию *DU*).

- Компонент оказывается в состоянии опасного необнаруженного отказа и переход Tr2 становится разрешенным.

- Обнаружение опасного отказа происходит после детерминированной задержки *d2* (*d2* = *t* modulo *τ*, где *t* – текущее время, *τ* – интервал проверки). Так моделируется проверочный интервал.

- Когда время *t2* выходит, т.е. опасный отказ обнаружен, метка переходит на позицию *DD*. Компонент сейчас ожидает ремонта и переход Tr3 становится разрешенным.

- Задержка *d3* для запуска перехода Tr3 (начала ремонта) не зависит от самого компонента, но готовность ресурсов для ремонта представляется сообщением *RA*. Это регулируется событием, происходящим из другой части всей сети Петри, не представленной на рисунке В.33.

- Ремонт начинается, как только у ремонтной бригады появляется готовность (т.е. ?RA = true становится истинным) и метка переходит в позицию *R*. Ремонтные ресурсы мгновенно становятся неготовыми для другого ремонта, и равенство !RA = false используется для обновления значения *RA*. Это предотвращает проведение еще одного ремонта в это же время.

- Случайный переход Tr4 (т.е. окончание ремонта) становится разрешенным и может быть рассчитана задержка *d4* при помощи случайного числа, соответствующего интенсивности ремонта *μ*.

- Когда проходит *d4*, запускается переход Tr4 и компонент вновь возвращается в нормальное рабочее состояние (метка переходит в позицию *W*). Ремонтные ресурсы вновь становятся готовыми и *RA* обновляется при помощи *!RA = true*.

- И так далее, до тех пор, пока запуск следующего разрешенного перехода попадает в заданный период [0, *T*].

Если следующий запуск уже не попадает в период [0, *T*], то моделирование останавливается и в результате для компонента формируется одна история. Во время формирования такой истории могут быть зафиксированы соответствующие параметры в виде их значений для маркируемых позиций (например отношение времени нахождения метки в конкретной позиции ко времени *T*), частоты запуска переходов, время до первого появления заданного события и т. д.

Идея метода Монте-Карло состоит в том, что формируется огромное количество таких историй и выполняется их статистическая обработка для получения адекватных параметров процесса.

В отличие от аналитических вычислений, метод Монте-Карло позволяет легко объединять детерминированные и случайные задержки, для которых может быть

выполнено моделирование на основе их кумулятивного распределения вероятностей $F(d)$ и случайных чисел z_i на отрезке [0, 1]. Такие случайные числа есть почти в любом языке программирования и разработаны мощные алгоритмы для подобного моделирования.

Затем случайная величина (d_i), распределенная в соответствии с $F(d)$, получается из случайной величины (z_i) при помощи операции: $d_i = F^{-1}(z_i)$. Это довольно просто, если существует аналитическое выражение для $F^{-1}(z)$, как, например, для экспоненциально распределенной задержки $d_i = -\frac{1}{\lambda_{DU}} Log(z_i)$.

Точность моделируемого параметра X обеспечивается статистическим анализом, который позволяет рассчитать среднее, дисперсию, стандартное отклонение и доверительный интервал моделируемого параметра:

- среднее: $\overline{X} = \frac{\Sigma_i x_i}{N}$;
- дисперсия: $\sigma^2 = \frac{\Sigma_i (x_i - \overline{X})^2}{N}$, и стандартное отклонение: σ;
- 90 %-ный доверительный интервал для $\overline{X}: Conf = 1{,}64 \frac{\sigma}{\sqrt{N}}$.

Таким образом, при использовании метода Монте-Карло всегда можно предсказать точность результатов. Например, 90 %-ная вероятность того, что истинный результат $\hat{X}$ принадлежит интервалу $\left[\overline{X} - 1{,}64\sigma/\sqrt{N},\quad \overline{X} + 1{,}64\sigma/\sqrt{N}\right]$.

Данный интервал уменьшается, когда количество историй возрастает и когда частота появления X растет.

На современных персональных компьютерах для Э/Э/ПЭ систем, связанных с безопасностью, несложно выполнить вычисления вплоть до УПБ 4.

B.5.3.3 Принцип расчета PFD

Подсеть Петри на рисунке B.33 можно непосредственно использовать для оценки PFD_{avg} компонента, потому что значение одного из параметров маркируемой позиции W, которое равно отношению времени нахождения метки в позиции W ко времени T, в действительности является средним значением готовности A компонента. В результате имеем: $PFD_{avg} = 1 - A$.

Точность вычислений, как было показано выше, можно оценить, используя статистический анализ.

Более сложное поведение можно представить, используя специальные подсети Петри. На рисунке B.34 показана идея, как можно выполнить моделирование

периодически проверяемых компонент, отказы по общей причине (*CCF*) и ремонтные ресурсы.

Слева представлена модель периодически проверяемого компонента, который переходит из состояния в состояние: рабочее состояние (*W*), опасный необнаруженный отказ (*DU*), проверка (*DUT*), опасный обнаруженный отказ (*DD*), готовность к ремонту (*RR*) и ремонт (*R*).

Когда происходит отказ (*DU*), формируется сообщение $!\text{-}C_i$ (эквивалентно $!C_i$ = *false*), чтобы сообщить, что компонент отказал. Затем он ждет, пока не запустится периодическая проверка (*DUT*). Интервал периодической проверки равен τ и смещение θ. После выполнения проверки в течение времени, равного π, система переходит в состояние *DD*. Если запасные части готовы (хотя бы одна метка в *SP*), то компонент становится готовым к ремонту (*RR*) и переменная *NbR* увеличивается на 1, чтобы проинформировать ремонтные ресурсы о количестве компонентов, которым необходим ремонт. После того, как ремонтные ресурсы доставлены к месту ремонта (одна метка в *OL*), начинается ремонт (*R*) и метка из *OL* удаляется. Когда вновь достигается нормально работающее состояние компонента, то формируется сообщение $!C_i$ (т.е. $!C_i$ = *true*), *NbR* уменьшается на единицу и метка возвращается обратно в *OL*, что разрешает выполнять дальнейшие ремонты. И так далее.

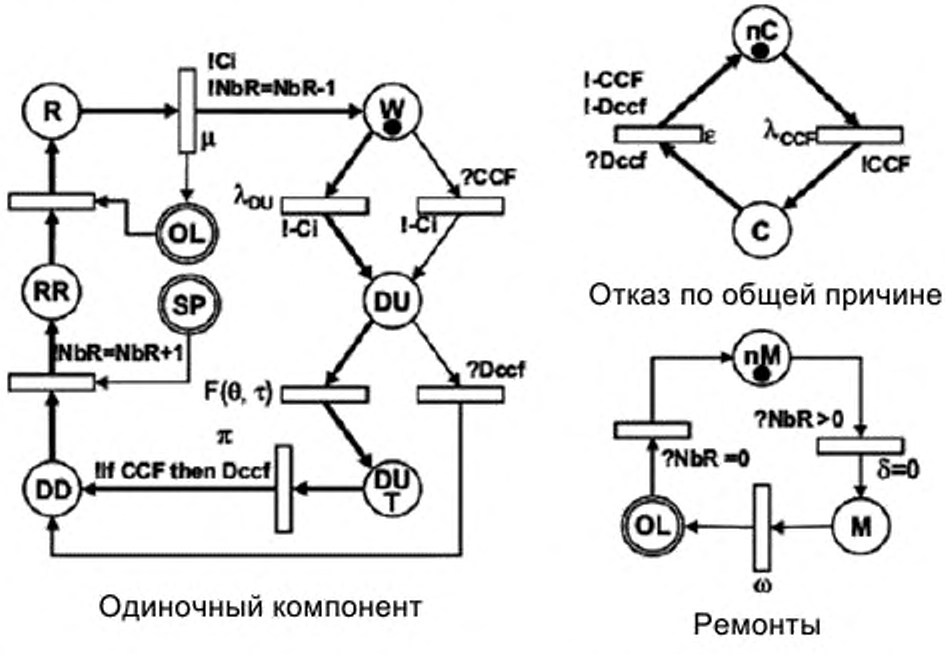


Рисунок В.34 – Сеть Петри, моделирующая отказ по общей причине и ремонтные

ресурсы

В подсетях Петри, моделирующих ремонт, используется переменная *NbR*. Когда ее величина становится больше нуля, начинается мобилизация ресурсов (*M*) и после определенной задержки они готовы выполнять работы на месте (*OL*). Метка в *OL* используется для проверки, начался ли ремонт одного из отказавших компонентов. Таким образом, только один ремонт может осуществляться в каждый момент времени. После выполнения всех ремонтов (т.е. *NbR* = 0) ремонтные ресурсы демобилизуются.

На рисунке В.34 также представлена модель отказа по общей причине (*CCF*). Когда происходит такой отказ (λ_{DCC}), сообщение *!CCF* становится истинным и используется для того, чтобы все отказавшие по общей причине компоненты перевести в их состояния *DU*. Соответствующее сообщение *Ci* становится ложным и компоненты ремонтируются независимо друг от друга. Когда проверка компонента завершена, утверждение *!IF CCF then* D_{ccf} позволяет сообщить всем остальным компонентам, что *CCF* был выявлен. Данное сообщение используется для того, чтобы незамедлительно перевести в их состояния *DD*. Это сообщение также используется для того, чтобы восстановить отказавшую по общей причине подсеть Петри, но это делается через некоторое время (ε), чтобы обеспечить, что все компоненты перед восстановлением были переведены в их состояние DD.

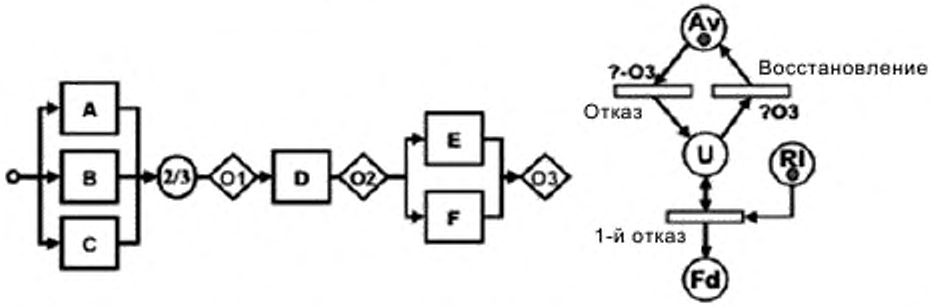


Рисунок В.35 – Использование блок-схемы надежности для построения сети Петри и вспомогательной сети Петри для вычислений *PFD* и *PFH*

Подсети Петри на рисунке В.34 используются как части более сложных моделей. Один из способов их использования показан на рисунке В.35, где представлена несколько адаптированная блок-схема надежности, заимствованная из рисунка В.16, куда добавлены промежуточные выводы O_i.

Для компонент A, B, C, D, E, F может быть выполнено моделирование с помощью набора подсетей Петри, представленных на рисунке В.34. Например, *CCF* для (A, B, C) и (E, F) с одними и теми же ремонтными ресурсами для всех компонентов. Остается только проблема связать компоненты вместе в соответствии с логикой блок-схемы надежности и расчета интересующего значения PFD_{avg}.

Связь компонентов можно очень легко выполнить при помощи сообщений C_i и построения следующих равенств:

$$O_1 = C_a C_b + C_a C_c + C_b C_c;$$

$$O_2 = O_1 C_d;$$

$$O_3 = O_2 \left(C_e + C_f\right).$$

Таким образом, когда O_3 истинно, вся Э/Э/ПЭ система, связанная с безопасностью, работает хорошо, иначе она не готова. Это сообщение использовано в подсети Петри с правой стороны для моделирования различных состояний Э/Э/ПЭ систем, связанных с безопасностью: готовность (*Av*), неготовность (*U*), безотказность (*Rl*) и состояние отказа (*Fd*).

Для расчета *PFD* важны лишь *Av* и *U*: когда O_3 становится ложным, система отказывает и становится неготовой; когда O_3 становится истинным, система восстановлена и становится опять готовой. Выполнить расчет достаточно просто, так как значение нахождения метки в состоянии *Av* является средним значением готовности системы, а значение нахождения метки в состоянии *U* является средним значением неготовности системы, то есть PFD_{avg}.

Таким образом, в методе Монте-Карло автоматически берется интеграл от мгновенной неготовности, но его ненужно вычислять за исключением случаев, для которых необходима зубчатая кривая. Это можно выполнить достаточно просто, оценив значение нахождения метки в состоянии *U* на всем периоде [0, *T*].

Описанное выше является иллюстрацией только основных направлений использования сетей Петри для вычисления УПБ, но потенциальные возможности моделирования безграничны.

В.5.3.4 Принцип расчета *PFH*

Для расчета *PFH* используются те же принципы, что указаны выше, и точно такие же подмодели могут быть использованы для *DU*-отказов. Рисунок В.36 представляет подсеть Петри, моделирующую отказ, который выявляется и ремонтируется, как только будет обнаружен.

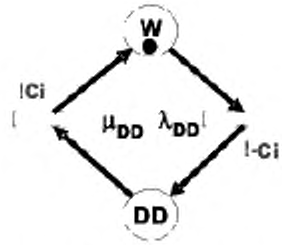


Рисунок В.36 – Простая сеть Петри для одного компонента с выявляемыми отказами и ремонтами

Как описано выше, такие модели компонентов могут быть использованы вместе с блок-схемами надежности, представляющими всю систему, как на рисунке В.35.

Если Э/Э/ПЭ система, связанная с безопасностью, работает в режиме с непрерывным запросом и является последним слоем безопасности, то инцидент происходит сразу после отказа, и *PFH* должно вычисляться через надежность системы. Это показано в нижней части подсети Петри, представленной справа на рисунке В.35. Средняя частота первого отказа системы на отрезке [0, *T*] является ее ненадежностью *F(T)*. Если значение *F(T)* достаточно мало по сравнению с 1, то в соответствии с определением *PFH* получаем: *PFH* = *F(T)/T*.

Ввиду того, что метка находится в *RI*, первый отказ является одним коротким переходом. При условии, что все истории ведут к отказу (т.е. период *T* достаточно длительный), среднее время нахождение метки в позиции *RI* является *MTTF* системы. Таким образом *PFH* ≈ 1/*MTTF* является верхней границей для *PFH*.

Если Э/Э/ПЭ система, связанная с безопасностью, работает в режиме с непрерывным запросом и не является последним слоем безопасности, то ее отказ непосредственно не ведет к инценденту. После полного отказа она ремонтируется и ее *PFH* должна вычисляться через неготовность системы. Эта величина получается непосредственно из частоты *Nbf* запуска перехода, моделирующего отказ. Таким образом получаем данные, сколько раз система отказала в течение указанного периода; в результате имеем *PFH(T)* = *Nbf*/*T*.

Если период *T* является достаточно большим, то *MUT* может быть рассчитан при помощи накопленного времени MCT_{Av} в состоянии *Av*, а *MDT* – при помощи среднего накопленного времени MCT_U в состоянии *U*. Средние накопленные времена MCT_A и MCT_U легко вычислить во время моделирования методом Монте-Карло, просто сложив время, когда метка находится в позициях *Av* или *U*. Получаем: *MUT* = MCT_A / *Nbf* и *MUT* =

MCT_u / *Nbf*. Это может быть использовано для вычисления $PFH = 1/(MUT + MDT) = 1/MTBF = Nbf/T$.

Все эти результаты получаются непосредственно, так как метод Монте-Карло легко определяет средние величины. Все описанное выше является лишь иллюстрацией широты использования сетей Петри для расчета УПБ, но реальные возможности моделирования являются практически безграничными.

B.5.4 Прочие подходы

Отношение между размером моделей и количеством компонентов изучаемой системы существенно изменяется в зависимости от используемого подхода. Для дерева отказов и сетей Петри - это линейное отношение, но для марковских процессов оно - экспоненциальное. Таким образом для моделирования сложных систем чаще используются деревья отказов и сети Петри, чем марковские процессы. По этой причине сети Петри иногда используются для создания больших марковских графов.

Формальные языки для описанных выше графических представлений формируют плоские модели: каждый элемент на каждом уровне описывается отдельно. Из-за этого большие модели иногда сложно осваивать и поддерживать. Одним из способов решения данной проблемы является использование структурного языка, описывающего компактную иерархическую модель. В последнее время был разработан ряд таких формальных языков, доступны также некоторые программные пакеты. В качестве примера можно рассмотреть язык AltaRica Data Flow, опубликованный в 2000 году для свободного использования сообществом по надежности и спроектированный для точного моделирования свойств корректно и некорректно функционирующих промышленных систем (см. B.7).

На рисунке B.37 показан эквивалент блок-схемы надежности, представленной на рисунке B.1. Данная модель является иерархической, потому что модели отдельных модулей созданы один раз и затем используются повторно по мере необходимости на разных уровнях моделирования системы. Это позволяет получать очень компактные модели.

В целях упрощения представления для компонентов отображены только два перехода: отказ и ремонт (т.е. *DD* отказы выявляются и исправляются по мере появления).

Логические операторы (or, and) используются для описания логики системы. Это сделано для прямой связи с блок-схемой надежности, и значение переменной *Out* моделирует состояние системы: если система в работоспособном состоянии, то

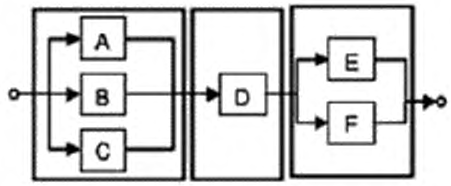
Out=true; если система в состоянии отказа, то *Out=false*.

Это позволяет создавать хорошие модели поведения для эффективного моделирования методом Монте-Карло, так что все описанное выше для PFD_{avg} и *PFH* остается верным и здесь. Поэтому далее эта тема не развивается.

Такой формальный язык обладает аналогичными математическими свойствами, что и сети Петри, и поэтому можно компилировать одну модель с другой без особого труда. Это также позволяет обобщать свойства языков дерева отказов и марковских процессов. Поэтому, если описание было ограничено свойствами марковских процессов или дерева отказов, то можно преобразовать модель в эквивалентные граф Маркова или дерево отказов. Ключевые слова «predicate» и «locker» в конце модели содержат указание на выполнение генерации дерева отказов или марковской модели или на применение метода Монте-Карло.

```
node bloc
  state working:bool;
  flow  input:bool:in; output:bool:Out;
  event failure, repair;
  init
    working:=true;
  trans
    working     |- failure -> working:=false;
    not working |- repair  -> working:=true;
  assert
    output = if working then input else false;
  extern
    law <event failure> = exponential(lambda);
    law <event repair>  = exponential(mu);
    parameter lambda = 1e-3;
    parameter mu     = 0.01;
edon

node main
  sub A:bloc; B:bloc;
      C:bloc; D:bloc;
      D:bloc; E:bloc;
  assert
    A.input = true, B.input = true,  C.input = true,
    D.input = (A.output or B.output  or C.output),
    E.input = D.output,
    F.input = D.output;
    Out     = (E.output or F.output)
  extern
    predicate failed  = <term (not Out)>;
    locker    failed  = <term (not Out)>;
edon
```



Отдельный модуль

Данные по надежности

Система

Нежелательное событие

Рисунок В.37 – Пример моделирования свойств корректно и некорректно функционирующих систем с использованием формального языка

Использование формального языка, созданного для моделирования поведения корректно и некорректно функционирующих систем, позволяет:

- выполнять моделирование методом Монте-Карло непосредственно на моделях;
- генерировать графы Маркова и выполнять аналитические вычисления, как показано ранее (когда язык ограничен марковскими свойствами);

- генерировать эквивалентное дерево отказов и проводить аналитические вычисления, как показано ранее (когда язык ограничен логическими свойствами).

Такие формальные языки, описывающие поведение корректно и некорректно функционирующих систем, являются языками общего назначения. Их без труда можно применять для конкретного анализа Э/Э/ПЭ систем, связанных с безопасностью. Эти языки предоставляют эффективный способ выполнения вычислений PFD_{avg} и *PFH* для Э/Э/ПЭ систем, связанных с безопасностью, с несколькими слоями защиты, различными типами режимов отказа, сложными структурами контрольных проверок, зависимостью компонентов, ресурсами обслуживания и т.д., т.е. когда прочие методы не подходят из-за своих ограничений.

B.6 Обработка неопределенностей

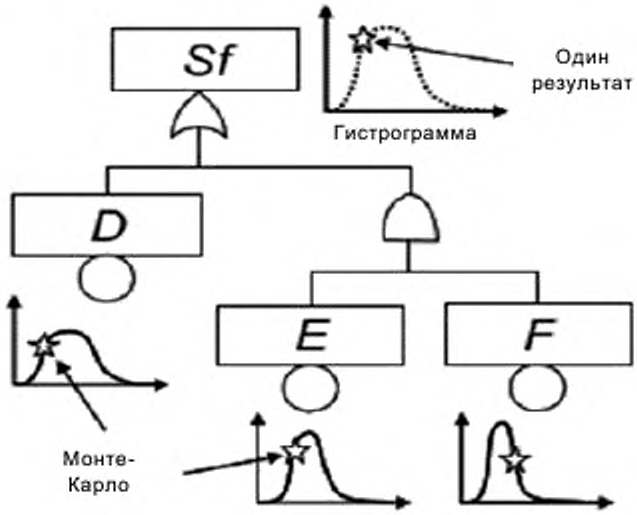


Рисунок B.38 – Принцип распространения неопределенности

Основная проблема, с которой сталкиваются при вероятностных расчетах, связана с неопределенностями в параметрах надежности. Таким образом, при проведении расчетов *PFD* и *PFH* полезно оценить, что и как влияет на неопределенность результатов.

К данной проблеме нужно подходить осторожно, но, как показано на рисунке B.38, использование метода Монте-Карло позволяет ее эффективно решать.

На данном рисунке входные параметры надежности (например, интенсивность

необнаруженных опасных отказов) более не являются определенными и поэтому заменены случайными переменными. Плотность вероятности таких случайных величин более или менее «острая» или «плоская» в зависимости от степени неопределенности: плотность вероятности *F* острее, чем *E* или *D*. Это означает, например, что неопределенность *F* меньше, чем *E* или *D*.

Порядок вычислений следующий:

1 Генерируется один набор входных параметров при помощи генератора случайных чисел в соответствии с вероятностным распределением данных параметров (аналогично тому, что описано в В.3.2).

2 Проводится одно вычисление, используя сгенерированный ранее набор входных параметров.

3 Записывается полученный результат (в него входит один результат, используемый на шаге 4).

4 Повторяются шаги с 1 по 3, пока не будет получено достаточное (например, 100 или 1000) количество значений для того, чтобы составить гистограмму (точечная линия на рисунке В.38).

5 Выполняется статистический анализ гистограммы для получения среднего значения и стандартного отклонения конечного результата.

Среднее значение гистограммы является PFD_{avg} или *PFH* в зависимости от выполненных вычислений, а стандартное отклонение определяет неопределенность результатов. Чем меньше стандартное отклонение, тем более точны вычисления PFD_{avg} или *PFH*.

Представленный выше порядок вычисления для дерева отказов является довольно общим и может быть применен к любому из методов, приведенных в настоящем приложении: упрощенные формулы, марковские процессы и даже сети Петри или формальные языки. Если вычисления по методу Монте-Карло уже проведены, то их нужно повторить.

Распределение вероятности для заданного входного параметра надежности должно быть выбрано в соответствии с собранным знанием о нем. Это может быть:

- равномерное распределение между верхней и нижней границами;
- треугольное распределение с наиболее вероятным значением;
- логонормальное распределение с заданным значением ошибки;
- распределение χ^2 (хи-квадрат) и т.д.

Первое можно оценить технически, когда данных реальных испытаний не очень много. Если данных реальных испытаний много, то можно использовать последнее распределение, так как данные реальных испытаний обеспечивают средние значения параметров, а также доверительные интервалы для этих средних значений.

Например, если наблюдается n отказов в течение накопленного времени наблюдения T, то имеем:

- $\bar{\lambda} = \frac{n}{T}$ является максимальным вероятным ожиданием интенсивности отказов;
- $\lambda_{Inf,\alpha} = \frac{1}{2T}\chi^2_{(1-\alpha),2n}$ нижняя граница с вероятностью α %, что будет ниже $\lambda_{Inf,\alpha}$;
- $\lambda_{Sup,\alpha} = \frac{1}{2T}\chi^2_{\alpha,2(n+1)}$ верхняя граница с вероятностью α %, что будет выше $\lambda_{Sup,\alpha}$.

Когда α = 5 %, истинное значение λ имеет 90 шансов из 100 принадлежать интервалу $[\lambda_{Inf,\alpha}, \lambda_{Sup,\alpha}]$. Чем меньше этот интервал, тем точнее значение λ. Обычно хорошие базы данных по надежности предоставляют эту информацию. Аналитики должны рассматривать данные по надежности, предоставляемые без доверительного интервала (или информации, позволяющей его рассчитать), очень осторожно.

могут быть использованы для построения подходящего распределения для моделирования интенсивности отказов λ заданного режима отказа и его неопределенностей. Это очевидно для распределения χ^2, но и для построения логонормального распределения было показано, что их использование также очень эффективно. Такой логонормальный закон определяется его средним $\bar{\lambda}$ или его медианой $\lambda_{50\%}$ и так называемым стандартным отклонением.

Это распределение имеет очень интересное свойство: $\lambda_{Inf,\alpha} = \frac{\lambda_{50\%}}{ef_\alpha}$ и $\lambda_{Sup,\alpha} = \lambda_{50\%}\, ef_\alpha$.

Тогда оно определяется только двумя параметрами: $\bar{\lambda}$ и $ef_\alpha \approx \sqrt{\frac{\lambda_{Sup,\alpha}}{\lambda_{Inf,\alpha}}}$.

Если ef_α = 1, то неопределенностей нет, если ef_α = 3.3, то верхняя и нижняя границы доверительного интервала различаются почти в 10 раз, и т. д..

Эти законы могут быть использованы, в свою очередь, вместе с методом Монте-Карло для того, чтобы учесть влияние средних значений и неопределенностей PFD_{avg} и PFH. Поэтому всегда возможно получить значение неопределенности с помощью вероятностных расчетов. Некоторые программные пакеты реализуют такие вычисления непосредственно.

При анализе избыточных систем анализ должен учитывать не только неопределенность интенсивности отказов основного элемента, но также и точность интенсивности *CCF*. Даже если существует хороший набор данных реальных испытаний для элементов, то редко имеется в наличии хороший набор данных реальных испытаний для *CCF* и, следовательно, это будет вносить наибольшую неопределенность.

B.7 Ссылки

См. в [3], [5] и [7] – [13] дополнительную информацию по вычислению вероятности отказа.

Приложение С
(справочное)

Расчет охвата диагностикой и доли безопасных отказов

Метод расчета охвата диагностикой и доли безопасных отказов приведен в МЭК 61508-2, приложение С. Настоящее приложение содержит краткое описание использования этого метода для расчета охвата диагностикой. Предполагается, что информация, представленная в МЭК 61508-2, доступна и при необходимости используется при получении значений, приведенных в таблице С.1. Возможные диапазоны охвата диагностикой для некоторых подсистем или компонент Э/Э/ПЭ систем, связанных с безопасностью, представлены в таблице С.2. Значения, представленные в таблице С.2, опираются на инженерные оценки.

Чтобы понять все значения таблицы С.1, потребовалась бы подробная схема аппаратных средств, с помощью которой можно определить влияние всех режимов отказов. Представленные в таблице С.1 значения приведены только в качестве примера (для некоторых компонентов таблицы С.1 охват диагностикой не определен, так как практически невозможно обнаружить все режимы отказов этих компонентов).

Таблица С.1 была сформирована следующим образом:

a) Для определения влияния каждого вида отказов каждого компонента на поведение системы без диагностических испытаний был проведен анализ видов и влияния отказов. Для каждого компонента приведены доли безопасных отказов S и опасных отказов D от общей интенсивности отказов, связанные с каждым видом отказов. Для простых компонентов деление на опасные и безопасные отказы может быть четко определено, в остальных случаях – основано на инженерной оценке. Для сложных компонентов, если детальный анализ каждого вида отказа невозможен, считают, что отказы делятся в соотношении: 50 % безопасных, 50 % – опасных. Для формирования таблицы С.1 использовались виды отказов, задаваемые именно таким распределением, хотя возможно и другое, более предпочтительное распределение по видам отказов.

b) Значения охвата диагностикой для каждого конкретного диагностического испытания каждого компонента помещают в столбце DC_{comp} таблицы С.1. В таблице С.1 также приведены конкретные значения охватов диагностикой для обнаружения как безопасных, так и опасных отказов. Было показано, что для простых компонентов (например резисторов, конденсаторов и транзисторов) отказы из-за отсутствия контакта или короткого замыкания обнаруживаются с охватом диагностикой 100 %, тем не менее

использование таблицы С.2 ограничивает охват диагностикой значением 90 % для компонента U16 комплексного компонента типа B.

c) В столбцах 1 и 2 таблицы С.1 приведены интенсивности безопасных λ_S и опасных λ_{DD} + λ_{DU} отказов для каждого компонента при отсутствии диагностических испытаний.

d) Обнаруженный опасный отказ считают фактически безопасным, что позволяет определить отношение между фактически безопасными отказами (т.е. любыми обнаруженными безопасными, необнаруженными безопасными или обнаруженными опасными отказами) и необнаруженными опасными отказами. Интенсивность фактически безопасных отказов определяют произведением значения интенсивности опасных отказов и значения охвата диагностикой для опасных отказов и сложением результата со значением интенсивности безопасных отказов (см. столбец 3 таблицы С.1). Точно так же интенсивность необнаруженных опасных отказов определяют вычитанием охвата диагностикой для опасных отказов из единицы и умножением результата на интенсивность опасных отказов (см. столбец 4 таблицы С.1).

e) В столбце 5 таблицы С.1 приведены значения интенсивности обнаруженных безопасных отказов, а в столбце 6 таблицы С.1 - значения интенсивности обнаруженных опасных отказов, полученные умножением значения охвата диагностикой на значения интенсивности безопасных и опасных отказов соответственно.

f) Использование таблицы С.1 дает следующие результаты:

– общая интенсивность безопасных отказов, включая обнаруженные опасные отказы:

$$\sum \lambda_S + \sum \lambda_{DD} = 9{,}9 \times 10^{-7};$$

– общая интенсивность необнаруженных опасных отказов:

$$\sum \lambda_{DU} = 5{,}1 \times 10^{-8};$$

– общая интенсивность отказов:

$$\sum \lambda_S + \sum \lambda_{DD} + \sum \lambda_{DU} = 1{,}0 \times 10^{-6};$$

– общая интенсивность необнаруженных безопасных отказов:

$$\sum \lambda_{SU} = 2{,}7 \times 10^{-8};$$

– охват диагностикой для безопасных отказов:

$$\frac{\sum \lambda_{SD}}{\sum \lambda_S} = \frac{3{,}38}{3{,}65} = 93\ \%;$$

– охват диагностикой для опасных отказов (обычно называемый «диагностическим охватом»):

$$\frac{\sum \lambda_{DD}}{\sum \lambda_{DD} + \sum \lambda_{DU}} = \frac{6{,}21}{6{,}72} = 92\ \%;$$

– доля безопасных отказов:

$$\frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_{DD} + \sum \lambda_{DU}} = \frac{986}{365 + 672} = 95\ \%.$$

g) Без диагностических испытаний интенсивность отказов распределяется следующим образом: 35 % безопасных отказов и 65 % – опасных отказов.

Таблица С.1 — Расчет охвата диагностикой и доли безопасных отказов

Компонент	№	Тип	Распределение на безопасные и опасные отказы для каждого вида отказов								Распределение на безопасные и опасные отказы для охвата диагностикой и рассчитанных интенсивностей отказов ($\times 10^{-9}$ ч$^{-1}$)						
			OC		*SC*		Изменение значения		Функциональные отказы		DC_{comp}		1	2	3	4	5	6
			S	*D*	*S*	*D*	*S*	*D*	*S*	*D*	*S*	*D*	λ_S	$\lambda_{DD}+\lambda_{DU}$	$\lambda_S+\lambda_{DD}$	λ_{DU}	λ_{SD}	λ_{DD}
Print	1	Печать	0,5	0,5	0,5	0,5	0	0	0	0	0,99	0,99	11,0	11,0	21,9	0,1	10,9	10,9
*CN*1	1	Con96pin	0,5	0,5	0,5	0,5					0,99	0,99	11,5	11,5	22,9	0,1	11,4	11,4
*C*1	1	100 нФ	1	0	1	0	0	0	0	0	1	0	3,2	0,0	3,2	0,0	3,2	0,0
*C*2	1	10 мкФ	0	0	1	0	0	0	0	0	1	0	0,8	0,0	0,8	0,0	0,8	0,0
*R*4	1	1 М	0,5	0,5	0,5	0,5					1	1	1,7	1,7	3,3	0,0	1,7	1,7
*R*6	1	100 К									0	0	0,0	0,0	0,0	0,0	0,0	0,0
*OSC*1	1	OSC24 МГц	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1	1	16,0	16,0	32,0	0,0	16,0	16,0
*U*8	1	74HCT85	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,99	0,99	22,8	22,8	45,4	0,2	22,6	22,6
*U*16	1	MC68000-12	0	1	0	1	0,5	0,5	0,5	0,5	0,90	0,90	260,4	483,6	695,6	48,4	234,4	435,2
*U*26	1	74HCT74	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,99	0,99	22,8	22,8	45,4	0,2	22,6	22,6
*U*27	1	74F74	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,99	0,99	14,4	14,4	28,7	0,1	14,3	14,3
*U*28	1	PAL16L8A	0	1	0	1	0	1	0	1	0,98	0,98	0,0	88,0	86,2	1,8	0,0	86,2
*T*1	1	BC817	0	0	0	0,67	0	0,5	0	0	1	1	0,0	0,2	0,4	0,0	0,0	0,2
Всего													365	672	986	50,9	338	621

S - безопасный отказ;
D - опасный отказ;
OC - потеря контакта;
SC - короткое замыкание;
DC_{comp} - охват диагностикой для компонента.

Примечания

1 Не обнаружен ни один вид отказа для компонента *R*6, т.е. его отказ не влияет на безопасность и готовность системы.

2 См. также таблицу В.1 (в настоящей таблице интенсивности отказов приведены только для отдельных рассматриваемых компонентов в канале, а не для каждого компонента).

Таблица С.2 – Уровни и диапазоны охвата диагностикой различных подсистем (компонентов)

Компонент	Низкий охват диагностикой	Средний охват диагностикой	Высокий охват диагностикой
Процессор (см. примечание 3):	в сумме менее 70 %	в сумме менее 90 %	-
- регистр	50 % - 70 %	85 % - 90 %	99 % - 99,99 %
- внутренняя регистровая память (см. примечание 3)	50 % - 60 %	75 % - 95 %	-
- блок кодирования и выполнения, включающий регистр тэгов (см. примечание 3)	50 % - 70 %	85 % - 98 %	-
- устройство вычисления адреса	50 % - 60 %	60 % - 90 %	85 % - 98 %
- счетчик команд	50 % - 70 %	-	-
- указатель стека	40 % - 60 %	-	-
Шина:			
- модуль управления памятью	50 %	70 %	90 % - 99 %
- устройство управления шины	50 %	70 %	90 % - 99 %
Обработка прерываний	40 % - 60 %	60 % - 90 %	85 % - 98 %
Кварцевый тактовый генератор (см. примечание 4)	50 %	-	95 % - 99 %
Контроль выполнения программы:			
- временное (см. примечание 3)	40 % - 60 %	60 % - 80 %	-
- логическое (см. примечание 3)	40 % - 60 %	60 % - 90 %	-
- временное и логическое (см. примечание 5)	-	65 % - 90 %	90 % - 98 %
Постоянная память	50 % - 70 %	99%	99,99 %
Непостоянная память	50 % - 70 %	85 % - 90 %	99 % - 99,99 %
Дискретное оборудование:			
- цифровой ввод/вывод	70 %	90 %	99 %
- аналоговый ввод/вывод	50 % - 60 %	70 % - 85 %	99 %
- источник питания	50 % - 60 %	70 % - 85 %	99 %
Устройство связи и запоминающее устройство большой емкости	90 %	99,9 %	99,99 %
Электромеханические устройства	90 %	99 %	99,9 %
Датчики	50 % - 70 %	70 % - 85 %	99 %
Оконечные элементы	50 % - 70 %	70 % - 85 %	99 %

Примечания

1 Настоящую таблицу применяют совместно с МЭК 61508-2 (таблица А.1, в которой приведены анализируемые виды отказов).

2 Если для охвата диагностикой задан конкретный диапазон,то верхние границы интервала могут быть определены только для узкого круга средств контроля или тестирования, которые реализуют чрезвычайно динамичную нагрузку для проверяемой функции.

3 В настоящее время для подсистем, схемы высокого охвата диагностикой которых отсутствуют, средства и методы высокой достоверности диагностики неизвестны.

4 В настоящее время для кварцевых тактовых генераторов средства и методы средней достоверности неизвестны.

5 Низкий диагностический охват для комбинации временного и логического контроля выполнения программы является средним.

Полезную информацию можно найти в [14] – [16].

Приложение D
(справочное)

Методика количественного определения влияния отказов аппаратных средств по общей причине в Э/Э/ПЭ системах

D.1 Общие положения

D.1.1 Введение

Настоящий стандарт включает в себя ряд методов, рассматривающих систематические отказы. Однако независимо от эффективности этих методов, существует остаточная вероятность возникновения систематических отказов. Это незначительно влияет на результаты расчета безотказности для одноканальных систем, однако возможность появления отказов, способных повлиять на несколько каналов многоканальной системы (или несколько компонентов в избыточной системе безопасности), т.е. отказов по общей причине, приводит к существенным ошибкам при расчетах безотказности многоканальных или избыточных систем.

В настоящем приложении приводится описание методики, позволяющей учитывать отказы по общей причине при оценке безопасности многоканальных или избыточных Э/Э/ПЭ систем. Использование данной методики дает более точную оценку полноты безопасности такой системы, чем при игнорировании отказов по общей причине.

Первая методика используется для расчета значения β-фактора, часто используемого при моделировании отказов по общей причине. Описываемая методика может быть использована для оценки интенсивности отказов по общей причине в случае двух или более параллельно работающих систем, если известна интенсивность случайных отказов аппаратных средств для одной из этих систем (см. D.5). Принято считать, что в общее число случайных отказов оборудования будет включено много отказов, которые были вызваны систематическими отказами.

В некоторых случаях предпочтительнее альтернативные методики, например, если благодаря наличию данных об отказах по общей причине можно получить более точное значение β-фактора или когда количество элементов, отказавших по общей причине, больше четырех. В качестве альтернативной методики, например, может быть использован метод биномиальной интенсивности отказов (также называемый «шоковая модель»).

D.1.2 Краткий обзор

Считается, что отказы в системе возникают из двух различных источников:

- случайные отказы аппаратных средств;
- систематические отказы.

Предполагается, что отказы первого вида возникают случайно по времени для любого компонента и приводят к отказу канала системы, частью которого является соответствующий компонент, тогда как отказы второго типа появляются сразу и детерминированным способом, когда система достигает состояния, в котором существует систематическая ошибка.

Существует некоторая вероятность того, что во всех каналах многоканальной системы могут произойти независимые случайные отказы аппаратных средств, вследствие чего все каналы одновременно окажутся неработоспособными. Так как предполагается, что такие отказы аппаратных средств возникают во времени случайно, вероятность таких отказов, одновременно возникающих в параллельных каналах, низка по сравнению с вероятностью отказа одного канала. Такая вероятность может быть рассчитана с помощью хорошо известных методов, но результат может быть очень оптимистичным, когда отказы не полностью независимы друг от друга.

Зависимые отказы обычно делятся на следующие группы [4]:

- Отказ по общей причине (CCF) вызывает несколько отказов по одной общей причине. Несколько отказов могут произойти одновременно или в течение некоторого периода времени;
- Отказы в общем режиме (CMF), которые являются частным случаем CCF, когда несколько единиц оборудования отказывают в одном режиме;
- Каскадные отказы, когда один компонент отказывает вследствие отказа другого компонента.

Термин CCF обычно используется для обобщения всех видов зависимых отказов, как сделано в настоящем приложении. Они также делятся на:

- зависимые отказы, вызванные понятными детерминированными причинами;
- события возможного остаточного многократного отказа, которые явно не анализируются из-за недостаточной точности их представления, отсутствия ясных детерминированных причин их возникновения или отсутствия возможности собрать данные по надежности.

Для первых должны быть выполнены анализ, моделирование и оценка общепринятым способом и только вторые должны быть обработаны, как показано в настоящем приложении D. Тем не менее, систематические отказы, которые являются полностью зависимыми отказами, не выявленными во время анализа безопасности

(иначе они должны быть устранены), обрабатываются определенным способом, указанным в настоящем стандарте, но данное приложение применяется в основном для случайных зависимых отказов аппаратных средств.

Таким образом, отказы по общей причине, являющиеся следствием одной причины, могут влиять на несколько каналов или несколько компонентов. Такая причина может быть следствием систематической ошибки (например конструктивной или ошибки технических условий) или внешнего воздействия, ведущего к преждевременным случайным аппаратным отказам (например избыточной температуры, возникающей из-за случайного отказа аппаратного средства, обычного вентилятора, что сокращает время жизни компонентов или нарушает заданные условия окружающей среды для их работы), или комбинации этих факторов. Так как отказы по общей причине чаще влияют на несколько каналов многоканальной системы, то вероятность такого отказа, скорее всего, будет доминирующим фактором при определении общей вероятности отказа многоканальной системы. Если не учитывать этот фактор, то будет трудно получить правильную оценку уровня полноты безопасности.

D.1.3 Защита от отказов по общей причине

Хотя отказы по общей причине являются следствием одной причины, они не обязательно проявляются во всех каналах одновременно. Например, при отказе вентилятора все каналы многоканальной Э/Э/ПЭ системы могут отказать, что ведет к отказу по общей причине. Однако необязательно все каналы нагреваются с одинаковой скоростью или имеют общую критическую температуру. Следовательно, отказы возникают в разных каналах в разное время.

Архитектура программируемых систем позволяет им выполнять внутреннее диагностическое тестирование непосредственно во время работы, что может быть реализовано различными способами, например:

- один канал ПЭ системы одновременно с обеспечением работы входного и выходного устройств может непрерывно выполнять внутреннюю проверку своей работы. На этапе проектирования можно достичь значения тестового охвата, равного 99 % [17]. Если 99 % внутренних сбоев обнаружены до того, как они приведут к отказу, вероятность сбоев одного канала, которые могут, в конечном счете, стать частью отказов по общей причине, значительно снижается;

- помимо внутреннего тестирования каждый канал ПЭ системы может отслеживать выходы других каналов многоканальной ПЭ системы (или каждое ПЭ-устройство может отслеживать другое ПЭ-устройство системы, состоящей из нескольких

ПЭ-устройств). Следовательно, отказ, возникший в одном канале, может быть обнаружен, и один или несколько оставшихся неотказавших каналов будут выполнять перекрестный контроль и инициировать безопасное выключение (следует отметить, что перекрестный контроль эффективен, если состояние системы управления постоянно меняется, например при наличии часто используемой в циклически работающем устройстве защитной блокировки или при внесении в устройство небольших изменений, не влияющих на управляющую функцию). Интенсивность выполняемого перекрестного контроля может быть достаточно высока, поэтому непосредственно перед неодновременными отказами по общей причине перекрестный контроль, скорее всего, обнаружит первый отказавший канал и позволит перевести систему в безопасное состояние до момента отказа второго канала.

Например для вентилятора скорость роста температуры и восприимчивость каналов несколько различаются, поэтому второй канал, возможно, откажет спустя несколько десятков минут после первого. Это позволяет после диагностического тестирования инициировать безопасное отключение первого отказавшего канала до того, как по общей причине откажет второй канал.

Таким образом:

- ПЭ системы обладают возможностью формировать барьеры защиты от отказов по общей причине и, следовательно, в меньшей степени подвержены им по сравнению с другими технологиями;

- для ПЭ систем можно использовать β-фактор, отличающийся от β-фактора для других технологий. Следовательно, оценки β-фактора, опирающиеся на предыдущие значения оценки интенсивности отказов, скорее всего, окажутся неправильными (ни одна из известных существующих моделей оценки вероятности отказа по общей причине не учитывает эффект автоматического перекрестного контроля);

- так как разнесенные во времени отказы по общей причине могут быть обнаружены с помощью диагностического тестирования до отказа всех каналов; подобные отказы могут не восприниматься как отказы по общей причине.

Рисунок D.1 – Связь между отказами по общей причине и отказами отдельных каналов

Существуют три способа уменьшения вероятности потенциально опасных отказов по общей причине:

1) уменьшение общего числа случайных аппаратных и систематических отказов (это уменьшает площади эллипсов, представленных на рисунке D.1, приводя к уменьшению площади пересечения эллипсов);

2) максимальное увеличение независимости каналов (это уменьшает площадь пересечения эллипсов, представленных на рисунке D.1, не меняя площади самих эллипсов);

3) обнаружение неодновременных отказов по общей причине, когда неисправным становится только один канал, до того как станет неисправным второй, т.е. использование диагностического тестирования или смещения контрольных проверок.

В системах более чем с двумя каналами, отказ по общей причине может повлиять на все каналы или только на несколько, но не на все, работающие в общем режиме. Таким образом, подход, представленный в данном приложении, в соответствии с первым методом, заключается в расчете значения β для дуплексной системы голосования 1oo2, а затем в использовании повышающего коэффициента для получаемого значения β в зависимости от общего количества каналов и схемы голосования (см. таблицу D.5).

D.1.4 Подход, адаптированный в серии стандартов МЭК 61508

Подход МЭК 61508 основан на выполнении следующих трех этапов:

1) использование методов по МЭК 61508-2/3 для снижения вероятности систематических отказов всей системы до уровня, соизмеримого с вероятностью случайных отказов аппаратных средств;

2) количественное определение факторов, которые могут быть определены количественно, т.е. учет вероятности случайных аппаратных отказов, как определено в МЭК 61508-2;

3) определение отношения, связывающего вероятность отказа по общей причине с вероятностью случайного отказа аппаратных средств с использованием практических средств, которые считаются лучшими в настоящее время. В настоящем приложении описана методика определения этого отношения.

Большинство методик оценки вероятности отказов по общей причине формируют прогнозы на основе вероятности случайного отказа аппаратных средств. Несомненно, непосредственной взаимосвязи между этими вероятностями нет, тем не менее, на практике некоторая корреляция между ними была найдена и, возможно, является следствием эффектов второго порядка. Например, высокая вероятность случайного отказа аппаратных средств системы связана:

- с большим объемом обслуживания, который требует система. А вероятность систематического отказа, являющегося следствием обслуживания, зависит от числа проведенных сеансов обслуживания, что также повышает интенсивность воздействия человеческих ошибок, приводящее к отказам по общей причине. Таким образом возникает связь между вероятностью случайного отказа аппаратных средств и вероятностью отказа по общей причине. Например:
  - после каждого случайного отказа аппаратных средств требуется ремонт, а за ним тестирование и, возможно, повторная калибровка;
  - для заданного уровня полноты безопасности система с большей вероятностью случайного отказа аппаратных средств требует более частого проведения контрольных проверок и с большей глубиной/сложностью, что также увеличивает влияние человеческого фактора;
- со сложностью системы. Вероятность случайного отказа аппаратных средств зависит от числа компонентов и, следовательно, сложности системы. Сложную систему труднее понять, поэтому у нее выше вероятность появления систематических ошибок. Кроме того, сложность системы затрудняет обнаружение отказов путем анализа или

тестирования и может приводить к тому, что часть логики системы будет выполняться только при редко встречающихся условиях. Это также приводит к появлению связи между вероятностью случайного отказа аппаратных средств и вероятностью отказа по общей причине.

В настоящее время используется несколько подходов для обработки CCF (β-фактор, несколько греческих букв, α-фактор, биномиальная интенсивность отказов и др.) [18]. Далее описаны две из текущих моделей, предлагаемые в настоящем приложении для третьего этапа трехэтапного подхода. Несмотря на ограничения, считается, что в настоящее время они представляют собой лучший способ обработки вероятности отказа по общей причине:

- устоявшаяся модель β-фактора, которая широко используется и обычно ее возможно использовать в многоканальных системах вплоть до четырех зависимых элементов;

- биномиальная интенсивность отказов [19] (также известная как шоковая модель), которая может быть использована, когда количество зависимых элементов больше четырех.

При использовании модели β-фактора для Э/Э/ПЭ системы возникают следующие две проблемы:

- выбор значения β-фактора. Многие источники (например [17]) предлагают диапазоны возможных значений β-фактора, но не определяют их конкретные значения, оставляя выбор за пользователем. Чтобы решить эту проблему, методика β-фактора, представленная в настоящем приложении, основывается на подходе, первоначально описанном в [20] и затем скорректированном в [21];

- ни модель β-фактора, ни шоковая модель не учитывают развитые возможности диагностического тестирования современных ПЭ систем, которыми можно воспользоваться для обнаружения неодновременных отказов по общей причине до того, как отказ полностью проявит себя. Для преодоления этой проблемы подход, описанный в [20] и скорректированный в [21], был изменен с тем, чтобы отразить влияние диагностического тестирования при оценке возможного значения β.

Функции диагностического тестирования, выполняющиеся внутри ПЭ системы, обеспечивают непрерывное сравнение работы ПЭ системы с заранее определенными состояниями. Эти состояния предварительно определяются программно или аппаратно (например с помощью контрольного таймера). Рассматриваемые таким образом функции диагностического тестирования можно считать дополнительными и частично

различающимися для каналов, работающих в ПЭ системе параллельно.

Также может использоваться метод перекрестного контроля каналов. Многие годы этот метод применялся в двухканальных системах с взаимной блокировкой, построенных исключительно на реле. Однако релейная технология обычно позволяет проводить перекрестное тестирование только во время изменения состояния каналов, что делает такое тестирование неподходящим для обнаружения неодновременных отказов по общей причине, если системы остаются в одном (например включенном) состоянии в течение длительного времени. С помощью технологии ПЭ системы перекрестный контроль может проводиться с высокой частотой.

D.2 Область применения методики

Область применения методики ограничена аппаратными отказами по общей причине по следующим причинам:

- модель β-фактора и шоковая модель связывают вероятность отказов по общей причине с вероятностью случайных аппаратных отказов. Вероятность отказов по общей причине, затрагивающих систему в целом, зависит от сложности системы (в которой главную роль, возможно, играет пользовательское программное обеспечение), а не только от аппаратуры. Очевидно, что любые расчеты, основанные на вероятности случайного аппаратного отказа, не могут учитывать сложность программного обеспечения;

- информирование об отказах по общей причине обычно ограничивается аппаратными отказами, что является главной заботой производителей оборудования;

- моделирование систематических отказов (например отказов программного обеспечения) считается практически неосуществимым;

- целью мероприятий, определенных в МЭК 61508-3, является снижение вероятности отказов по общей причине, связанных с программным обеспечением, до значения, приемлемого для необходимого уровня полноты безопасности.

Следовательно, оценка вероятности отказа по общей причине, выполненная по данной методике, связана только с аппаратными отказами. Эту методику не допускается использовать для получения интенсивности отказов всей системы, учитывающей вероятность отказа, связанную с программным обеспечением.

D.3 Особенности методики

Так как на датчики, логическую подсистему и исполнительные элементы влияют, например различные условия окружающей среды и диагностические тесты с разным уровнем возможностей, для каждой из этих подсистем настоящую методику применяют

независимо. Например логическую подсистему проще поместить в контролируемую среду, а датчики могут быть установлены снаружи и подвергнуты внешнему воздействию.

Программируемые электронные каналы предоставляют возможность для реализации разнообразных функций диагностического тестирования и способны:

- обеспечивать высокий охват диагностикой в пределах конкретных каналов;
- контролировать дополнительные избыточные каналы;
- обеспечивать высокую частоту повторения;
- контролировать с повышенной частотой датчики и/или исполнительные элементы.

Чаще всего отказы по общей причине не возникают одновременно во всех затронутых каналах. Поэтому, если частота повторения диагностического тестирования достаточно высока, большую часть отказов по общей причине можно обнаружить и, следовательно, устранить до того, как будут затронуты остальные доступные каналы.

Не все функции многоканальной системы, обеспечивающие устойчивость к отказам по общей причине, можно проверить с помощью диагностического тестирования. Однако эффективность этих функций, связанных с диверсификацией или независимостью, постоянно повышается. Любая функция, которая, возможно, увеличивает время между отказами каналов в случае неодновременного отказа по общей причине (или уменьшает долю одновременных отказов по общей причине), увеличивает вероятность обнаружения отказа при диагностическом тестировании и перевода установки в безопасное состояние. Следовательно, функции, связанные с устойчивостью к отказам по общей причине, делятся на функции, влияние которых предположительно возрастает при использовании диагностического тестирования, и влияние которых не меняется (см. таблицу D.1, столбцы X и Y соответственно).

Хотя для трехканальной системы вероятность отказов по общей причине, влияющих на все три канала, скорее всего значительно ниже вероятности отказов, влияющих на два канала, для упрощения методики β-фактора предполагается, что вероятность отказов не зависит от числа затрагиваемых каналов, т.е. возникающий отказ по общей причине затрагивает все каналы. Альтернативой является шоковая модель.

Данных об аппаратных отказах по общей причине, необходимых для калибровки методики, не существует, поэтому данные таблиц в настоящем приложении основываются на инженерных оценках.

Иногда процедуры диагностического тестирования не рассматриваются как необходимые для обеспечения безопасности, поэтому их уровень обеспечения качества может быть ниже, чем процедур, обеспечивающих основные функции управления. Данная методика была разработана в предположении, что уровень полноты безопасности для диагностического тестирования соответствует требуемому. Следовательно, любые программные процедуры диагностического тестирования должны разрабатываться с использованием методов, соответствующих требуемому уровню полноты безопасности.

D.4 Использование β-фактора для вычисления вероятности отказа Э/Э/ПЭ системы, связанной с безопасностью, из-за отказов по общей причине

Влияние отказов по общей причине на многоканальную систему с диагностическим тестированием следует рассматривать в каждом из каналов системы.

Используя модель β-фактора, для интенсивности опасных отказов по общей причине получим $\lambda_D\beta$, где λ_D –интенсивность опасных случайных отказов аппаратных средств для каждого отдельного канала, а β - β-фактор в отсутствие диагностического тестирования, т.е. доля отказов одного канала, влияющих на все каналы.

Предположим, что отказы по общей причине влияют на все каналы, а промежуток времени между таким влиянием на первый и остальные каналы мал по сравнению с интервалом времени между последовательными отказами каналов по общей причине.

Пусть в каждом канале применяется диагностическое тестирование, которое обнаруживает и вскрывает часть отказов. Отказы подразделяют на две категории: отказы, которые находятся вне охвата диагностического тестирования (т.е. никогда не могут быть обнаружены), и отказы в пределах охвата (которые, в конечном счете, будут обнаружены диагностическим тестированием).

Поэтому общую интенсивность отказов системы, вызванных опасными отказами по общей причине, вычисляют по формуле

$$\lambda_{DU}\beta + \lambda_{DD}\beta_D,$$

где λ_{DU} – интенсивность необнаруженных отказов одного канала, т.е., интенсивность опасных отказов, находящихся за пределами охвата диагностического тестирования; очевидно, любое уменьшение β - фактора, являющееся следствием частоты проведения диагностического тестирования, не может повлиять на λ_{DU};

β – фактор отказов по общей причине для необнаруживаемых опасных отказов, который равен общему β-фактору, применяемому в отсутствие диагностического тестирования;

λ_{DD} – интенсивность обнаруженных опасных отказов одного канала (т.е. интенсивность опасных отказов одного канала), находящихся в области охвата диагностического тестирования; если частота проведения диагностического тестирования высока, доля обнаруженных отказов ведет к уменьшению значения β, т.е. β_D;

β_D – доля опасных отказов по общей причине, обнаруживаемых диагностическими тестами. С увеличением частоты проведения диагностического тестирования значение β_D становится меньше β.

Значение β определяется по таблице D.5, которая использует результаты D.4, с помощью оценки $S = X + Y$ (см. D.5).

Значение β_D определяется по таблице D.5, которая использует результаты D.4, с помощью оценки $S_D = X(Z+1)+Y$.

D.5 Использование таблиц для оценки β

Оценку β-фактора рассчитывают отдельно для датчиков, логической подсистемы и исполнительных элементов.

Чтобы свести к минимуму вероятность возникновения отказов по общей причине, следует сначала определить средства, эффективно защищающие от появления таких отказов. Реализация соответствующих средств в системе ведет к уменьшению значения β-фактора, используемого при оценке вероятности отказа системы из-за отказов по общей причине.

Мероприятия и соответствующие им значения (баллы) параметров X и Y, полученные с помощью инженерной оценки и описывающие вклад каждого из мероприятий в уменьшение числа отказов по общей причине, перечислены в таблице D.1. Так как датчики и исполнительные элементы анализируются иначе, чем программируемая электроника, в таблице D.1 используются столбцы X_{LS} и Y_{LS} для программируемых электронных средств и столбцы X_{SF} и Y_{SF} для датчиков или исполнительных элементов.

Программируемые электронные системы могут использовать интенсивное диагностическое тестирование, позволяющее обнаруживать неодновременные отказы по общей причине. Для учета диагностического тестирования при оценке β-фактора общий вклад каждого из мероприятий в таблице D.1 разделен с использованием инженерной оценки на наборы значений X и Y. Для каждого конкретного мероприятия отношение $X:Y$ представляет собой меру повышения вклада этого мероприятия в борьбу с отказами по общей причине благодаря диагностическому тестированию.

Пользователь таблицы D.1 должен определить, какие мероприятия будут использованы для рассматриваемой системы, и сложить соответствующие мероприятиям баллы, приведенные в графах X_{LS} и Y_{LS} для логической подсистемы, или в графах X_{SF} и Y_{SF} – для датчиков или исполнительных элементов, получив суммы X и Y соответственно.

Коэффициент Z определяют по таблицам D.2 и D.3 по частоте и охвату диагностического тестирования с учетом примечания 4, определяющего, когда следует использовать ненулевое значение Z. Затем (при необходимости) рассчитывают сумму баллов S (см. D.5) по формуле $S = X + Y$ - для получения значения β (β-фактора для необнаруженных отказов) и $S_D = X(Z + 1) + Y$ - для получения значения β_{Dint} (β_D - фактора для обнаруженных отказов), где S или S_D – баллы, используемые в таблице D.4 для определения соответствующего β_{int} - фактора.

β_{int} и β_{Dint} являются значениями отказа по общей причине до рассмотрения эффекта различных степеней избыточности.

Таблица D.1 – Оценка мероприятий защиты программируемых электронных средств или датчиков/исполнительных элементов от возникновения отказов по общей причине

Мероприятие	Логическая подсистема		Датчики и исполнительные элементы	
	X_{LS}	Y_{LS}	X_{SF}	Y_{SF}
Разделение/выделение				
Везде ли сигнальные кабели каналов разделены между собой	1,5	1,5	1,0	2,0
Расположены ли логические подсистемы каналов на отдельных печатных платах	3,0	1,0	-	-
Расположены ли логические подсистемы каналов в отдельных шкафах	2,5	0,5	-	-
Если датчики/исполнительные элементы включают в себя собственную управляющую электронику, то расположена ли электроника для каждого канала на отдельной печатной плате	-	-	2,5	1,5
Если датчики/ исполнительные элементы включают собственную управляющую электронику, то расположена ли электроника для каждого канала в различных помещениях и различных шкафах	-	-	2,5	0,5
Диверсификация/избыточность				
Реализованы ли в каналах различные электрические технологии, например, один канал электронный или программируемый электронный, а для другого используются реле	8,0	-	-	-
Реализованы ли в каналах различные электронные технологии, например, один канал - электронный, а другой - программируемый электронный	6,0	-	-	-
Используют ли устройства различные физические принципы для датчиков, например, давления и температуры, анемометр с вертушкой и доплеровский датчик и т.д.	-	-	9,0	-
Используют ли устройства различные электрические принципы/конструкции, например, цифровые и аналоговые, с компонентами от различных производителей (но не уцененные) или с различной технологией	-	-	6,5	-
Применяется ли низкая диверсификация, например, диагностическое тестирование аппаратуры использует одинаковую технологию	2,0	1,0	-	-
Применяется ли средняя диверсификация, например, диагностическое тестирование аппаратуры использует различную технологию	3,0	2,0	-	-
Были ли разработаны каналы различными конструкторами, которые не взаимодействовали между собой в процессе разработки	1,5	1,5	-	-
Использовались ли для каждого канала различные люди и различные методы тестирования в процессе его пуска	1,0	0,5	1,0	1,0
Обслуживается ли каждый канал в разное время разными людьми	3,0	-	3,0	-
Сложность/конструкция/применение/завершенность/опыт				
Предотвращает ли перекрестная связь между каналами обмен любой информацией, кроме используемой для диагностического тестирования или голосования	0,5	0,5	0,5	0,5
Превышает ли время использования в отрасли методов, применяемых для проектирования аппаратуры, пять лет	0,5	1,0	1,0	1,0
Превышает ли время работы с этим же оборудованием в аналогичных условиях пять лет	1,0	1,5	1,5	1,5
Проста ли система, например, имеет ли она не более 10 входов или выходов на канал	-	1,0	-	-
Защищены ли входы и выходы от возможного превышения безопасных значений напряжения и тока	1,5	0,5	1,5	0,5
С запасом ли рассчитаны все устройства/компоненты (например с коэффициентом 2 или более)	2,0	-	2,0	-

Продолжение таблицы D.1

Мероприятие	Логическая подсистема		Датчики и исполнительные элементы	
	X_{LS}	Y_{LS}	X_{SF}	Y_{SF}
Оценка/анализ и обратная связь				
Были ли изучены результаты анализа видов и влияния отказов или дерево неисправностей для того, чтобы установить источники отказов по общей причине, и устранены ли при проектировании предварительно известные источники отказов по общей причине	-	3,0	-	3,0
Рассматривались ли отказы по общей причине при анализе проекта при последующем внесении изменений в проект (требуется документальное доказательство действий по анализу проекта)	-	3,0	-	3,0
Все ли возможные отказы были полностью проанализированы и учтены в проекте (требуется документальное доказательство процедуры)	0,5	3,5	0,5	3,5
Процедуры/интерфейс пользователя				
Существует ли зафиксированная письменно схема работы, гарантирующая обнаружение отказов (или ухудшение характеристик) всех компонентов, установление корневых причин и проверку других аналогичных вопросов для аналогичных возможных причин отказов	-	1,5	0,5	1,5
Предусмотрены ли процедуры, обеспечивающие: разнесение обслуживания (включая настройку или калибровку) по времени любой части независимых каналов; возможность выполнения диагностического тестирования помимо ручных проверок, проводимых в ходе очередного обслуживания, между завершением обслуживания одного канала и началом обслуживания другого	1,5	0,5	2,0	1,0
Определено ли в документированных процедурах обслуживания, что обеспечивающие избыточность компоненты систем (например кабели и.т.д.) должны быть независимы друг от друга и закреплены в устройстве	0,5	0,5	0,5	0,5
Проводится ли обслуживание печатных плат и т.д. вне рабочего места (в центре ремонта компонентов) и проводится ли тестирование отремонтированных элементов перед их установкой	0,5	1,0	0,5	1,5
Обеспечивает ли система низкий охват диагностикой (от 60 % до 90 %) и сообщает ли об отказах на уровне модуля, допускающего замену в процессе эксплуатации	0,5	-	-	-
Обеспечивает ли система средний охват диагностикой (от 90 % до 99 %) и сообщает ли об отказах на уровне модуля, допускающего замену в процессе эксплуатации	1,5	1,0	-	-
Обеспечивает ли система высокий охват диагностикой (> 99 %) и сообщает ли об отказах на уровне модуля, допускающего замену в процессе эксплуатации	2,5	1,5	-	-
Сообщает ли диагностическое тестирование системы об отказах на уровне модуля, допускающего замену в процессе эксплуатации	-	-	1,0	1,0
Компетентность/обучение/культура безопасности				
Обучены ли конструкторы (с помощью обучающей документации) понимать причины и следствия отказов по общей причине	2,0	3,0	2,0	3,0
Обучен ли обслуживающий персонал (с помощью обучающей документации) понимать причины и следствия отказов по общей причине	0,5	4,5	0,5	4,5
Контроль состояния окружающей среды				
Ограничен ли доступ персонала (закрытые шкафы, недоступное размещение компонентов и т.д.)	0,5	2,5	0,5	2,5
Возможно ли, что система всегда будет работать в заданных диапазонах температур, влажности, коррозии, пыли, вибрации и т.д., в которых ее работа была проверена, без использования внешнего контроля состояния окружающей среды	3,0	1,0	3,0	1,0

Мероприятие	Логическая подсистема		Датчики и исполнительные элементы	
	X_{LS}	Y_{LS}	X_{SF}	Y_{SF}
Все ли сигнальные и силовые кабели отделены друг от друга	2,0	1,0	2,0	1,0

Окончание таблицы D.1

Мероприятие	Логическая подсистема		Датчики и исполнительные элементы	
	X_{LS}	Y_{LS}	X_{SF}	Y_{SF}
Проверка влияния окружающей среды				
Было ли проверено, что система устойчива ко всем воздействиям окружающей среды (например ЭМС, температура, вибрация, ударные нагрузки, влажность) на уровне, заданном в соответствующих международных или национальных стандартах	10,0	10,0	10,0	10,0

Примечания

1 Ряд факторов, связанных с работой системы, трудно предсказать во время проектирования. В таких случаях конструкторы должны убедиться в том, что конечный пользователь системы уведомлен, например, о процедурах, используемых для достижения требуемого уровня полноты безопасности. Необходимая информация должна быть включена в сопроводительную документацию.

2 Значения *X* и *Y* основаны на инженерной оценке и учитывают как косвенное, так и прямое влияние мероприятий. Например, использование модулей, допускающих замену во время эксплуатации, приводит:

- к выполнению ремонтных работ производителем в соответствующих условиях вместо ремонтных работ, выполняемых на месте в менее подходящих условиях. Это вносит свой вклад в значения *Y*, так как снижается вероятность систематических отказов и, следовательно, отказов по общей причине;
- к снижению необходимости вмешательства человека на месте и к возможности быстрой замены неисправных модулей, не выключая системы, повышая таким образом эффективность диагностики для идентификации отказов до того, как они станут отказами по общей причине. Это заметно влияет на значения *X*.

Таблица D.2 — Значение *Z*: программируемая электроника

Диагностический охват	Периодичность диагностического тестирования		
	Менее 1 мин	От 1 до 5 мин	Более 5 мин
≥ 99 %	2,0	1,0	0
≥ 90 %	1,5	0,5	0
≥ 60 %	1,0	0	0

Таблица D.3 — Значение *Z*: датчики или исполнительные элементы

Диагностический охват	Периодичность диагностического тестирования			
	Менее 2 ч	От 2 ч до 2 дней	От 2 до 7 дней	Более 7 дней
≥ 99 %	2,0	1,5	1,0	0
≥ 90 %	1,5	1,0	0,5	0
≥ 60 %	1,0	0,5	0	0

Примечания

1 Данная методика наиболее эффективна, если при подсчете баллов равномерно учитываются все группы мероприятий, представленные в таблице D.1. Следовательно, рекомендуется, чтобы общая сумма баллов *X* и *Y* для каждой группы была не менее общей суммы баллов *X* и *Y*, деленной на 20. Например, если общая сумма баллов *X*+*Y* равна 80, то общая сумма баллов *X*+*Y* для любой из групп (например для группы мероприятий «Процедуры/интерфейс пользователя») должна быть не менее четырех.

2 При использовании таблицы D.1 следует учитывать баллы для всех реализованных в системе мероприятий. Подсчет суммы баллов был разработан для учета тех мероприятий, которые не являются взаимно исключающими. Например, для системы, логические подсистемы каналов которой расположены в отдельных стойках, подсчитывают сумму баллов мероприятий таблицы D.1 "Расположены ли логические подсистемы каналов в отдельных шкафах" и "Расположены ли логические подсистемы каналов на отдельных печатных платах".

3 Если в датчиках или исполнительных элементах используется программируемая электроника, их рассматривают как часть логической подсистемы, если они находятся в том же здании (транспортном средстве), что и устройство, являющееся главной частью логической подсистемы, и в качестве датчиков или исполнительных элементов, если они расположены отдельно.

4 Для того, чтобы использовать ненулевое значение *Z*, нужно убедиться, что управляемое оборудование переходит в безопасное состояние до того, как неодновременный отказ по общей причине сможет повлиять на все каналы. Время, необходимое для обеспечения этого безопасного состояния, должно быть менее заявленного интервала диагностического тестирования. Ненулевое значение *Z* допускается использовать только в случае, если:

- система инициирует автоматическое выключение при обнаружении сбоя, или
- безопасное выключение не инициируется после первого сбоя[1)], но диагностическое тестирование:
  - определяет местонахождение сбоя и может его локализовать, а также
  - сохраняет способность перевода УО в безопасное состояние после обнаружения любых последующих сбоев, или
- применяется формальная система работы, гарантирующая, что причина любого обнаруженного сбоя будет полностью проанализирована в течение заявленного периода диагностического тестирования и либо:
  - установка немедленно выключается, если сбой может привести к отказу по общей причине, либо

[1)] Необходимо учитывать действия системы при обнаружении сбоя. Например, простая система с архитектурой голосования 2oo3 должна быть выключена (или отремонтирована) после обнаружения одиночного отказа в течение времени, приведенного в таблице D.2 или D.3. Если система не выключена, отказ второго канала может привести к тому, что при голосовании два отказавших канала получат перевес голосов над оставшимся (работоспособным) каналом. У системы, которая автоматически сама меняет архитектуру голосования на 1oo2 при отказе одного канала и автоматически выключается при возникновении второго отказа, вероятность обнаружения неисправности второго канала повышается и, следовательно, ненулевое значение *Z* возможно.

- канал, в котором произошел сбой, восстанавливается в течение заявленного интервала диагностического тестирования.

5 В обрабатывающих отраслях вряд ли возможно выключать УО при обнаружении сбоя во время интервала диагностического тестирования в соответствии с таблицей D.2. Настоящая методика не должна восприниматься как содержащая строгое требование выключать технологические установки непрерывного производства при обнаружении подобных сбоев. Однако если выключение не производится, то уменьшить β-фактор с помощью использования диагностического тестирования для программируемых электронных средств невозможно. В ряде других отраслей выключение УО во время интервала диагностического тестирования возможно. В этих случаях допускается использовать ненулевое значение *Z*.

6 Если диагностическое тестирование проводится модульно, то время повторения, приведенное в таблице D.2 или D.3, - это время между завершениями последовательного диагностического тестирования всего набора модулей. Охват диагностикой – общий охват, обеспечиваемый всеми модулями.

Таблица D.4 — Расчет величины β или β_D

Баллы (S или S_D)	Значение β или β_D для	
	логической подсистемы	датчиков или исполнительных элементов
120 или более	0,5 %	1 %
От 70 до 120	1 %	2 %
От 45 до 70	2 %	5 %
Менее 45	5 %	10 %

Примечания

1 Максимальные уровни β_D ниже обычно используемых, что объясняется использованием методов, описанных в настоящем стандарте, для уменьшения вероятности систематических отказов в целом и в результате - вероятности отказов по общей причине.

2 Значения β_D менее 0,5 % для логической подсистемы и 1 % - для датчиков трудно подтвердить.

Значение β_{int}, полученное из таблицы D.4, является отказом по общей причине, сопоставленным с системой 1oo2. Для других уровней избыточности (MooN) значение β_{int} меняется, как показано в таблице D.5, для получения окончательного значения β.

Таблица D.5 также используется для определения конечного значения β_D, но, когда есть β_{int}, оно может быть уменьшено до $\beta_{D\ int}$.

Примечание - Для дополнительной актуальной информации (по PDS методам) см [22].

Таблица D.5 — Расчет β для систем с уровнем резервирования, большим 1oo2

MooN				
	2	3	4	5

M	1	β_{int}	0,5 β_{int}	0,3 β_{int}	0,2 β_{int}
	2	-	1,5 β_{int}	0,6 β_{int}	0,4 β_{int}
	3	-	-	1,75 β_{int}	0,8 β_{int}
	4	-	-	-	2 β_{int}

D.6 Использование методики

Для демонстрации результата использования методики значения β и β_D для программируемых электронных средств приведены в таблице D.6.

Для всех групп мероприятий, кроме – «диверсификация/избыточность», были использованы типовые значения *X* и *Y*. Они были получены делением максимального значения баллов для конкретных групп на два.

Для систем с разнообразием значения *X* и *Y* для группы «диверсификация/избыточность» были выведены исходя из следующих мероприятий, рассмотренных в таблице D.1:

- одна система электронная, другая использует технологию реле;
- диагностическое тестирование аппаратных средств использует различные технологии;
- разные конструкторы не взаимодействовали между собой в процессе проектирования;
- для пуска системы использовались различные методы тестирования и разный персонал;
- обслуживание проводилось в разное время разными людьми.

Для систем с избыточностью значения *X* и *Y* для группы «диверсификация/избыточность» были выведены, исходя из того, что диагностика аппаратных средств проводилась независимой системой, использующей ту же технологию, что и системы с избыточностью.

В системах с разнообразием и в системах с избыточностью для величины *Z* использовались максимальные и минимальные значения, поэтому в таблице D.6 значения β и β_D представлены для четырех систем.

Таблица D.6 — Значения β и β_D для программируемых электронных средств

Группа мероприятий		Система с разнообразием и хорошим диагностическим тестированием	Система с разнообразием и плохим диагностическим тестированием	Система с избыточностью и хорошим диагностическим тестированием	Система с избыточностью и плохим диагностическим тестированием
Разделение/выделение	X	3,50	3,50	3,50	3,50
	Y	1,50	1,50	1,50	1,50
Диверсификация/избыточность	X	14,50	14,50	2,00	2,00
	Y	3,00	3,00	1,00	1,00
Сложность/конструкция/..	X	2,75	2,75	2,75	2,75
	Y	2,25	2,25	2,25	2,25
Оценка/анализ/....	X	0,25	0,25	0,25	0,25
	Y	4,75	4,75	4,75	4,75
Процедуры/интерфейс пользователя	X	3,50	3,50	3,50	3,50
	Y	3,00	3,00	3,00	3,00
Компетентность/обучение/...	X	1,25	1,25	1,25	1,25
	Y	3,75	3,75	3,75	3,75
Контроль состояния окружающей среды	X	2,75	2,75	2,75	2,75
	Y	2,25	2,25	2,25	2,25
Проверка влияния окружающей среды	X	5,00	5,00	5,00	5,00
	Y	5,00	5,00	5,00	5,00
Охват диагностикой	Z	2,00	0,00	2,00	0,00
X (всего)		33,5	33,5	21	21
Y (всего)		25,5	25,5	23,5	23,5
Сумма баллов S		59	59	44,5	44,5
β		2 %	2 %	5 %	5 %
Сумма баллов S_D		126	59	86,5	44,5
β_D		0,5 %	2 %	1 %	5 %

D.7 Биномиальная интенсивность отказов (шоковая модель) – подход CCF

Практические испытания отказов по общей причине (CCF) показывают, что если случается много двойных отказов и мало тройных, возможно, один четырехкратный, и ни

одного большего порядка при наблюдении за одной явной причиной, которая не могла быть определена во время анализа безопасности, то, следовательно, вероятность множественных отказов уменьшается с увеличением порядка CCF. Поэтому, если модель β-фактора является реалистичной для двойного отказа и немного пессимистичной для тройного, то для четырехкратного отказа и дальше она становится уж слишком консервативной. Рассмотрим типичный пример инструментальной системы безопасности, которая закрывает n скважин (например, n = 150) на нефтяном месторождении, когда происходит блокировка выхода. Конечно, 2, 3 или 4 скважины могут не закрыться из-за неявного CCF, но не n, как было смоделировано по β-фактору (иначе CCF будет явным и должны анализироваться отдельные отказы). Другой типичный пример возникает при работе с несколькими слоями безопасности в одно и то же время. Рассмотрение, например, возможных CCF между датчиками двух слоев безопасности может означать рассмотрение CCF между шестью датчиками (т.е. тремя датчиками на каждый слой).

Чтобы справиться с этой трудностью, было предложено несколько моделей [4], но большинство из них требуют достаточно много параметров надежности (например, множественные греческие буквы или α-модели), что становятся нереальными. Среди них биномиальная интенсивность отказов (шоковая модель), введенная в 1977 году Весели (Vesely) и улучшенная в 1986 Этвудом (Atwood), предусматривает более прагматичное решение [4, 6]. Идея в том, что когда происходит CCF, это похоже на удар по связанным компонентам. Этот удар может быть летальным (т.е. такое же влияние, как и в модели β-фактора) или нелетальным, и в этом случае есть только определенная вероятность, что данный компонент откажет из-за удара. Тогда вероятность того, что из-за удара будет получено k отказов, распределена биномиально.

Данной модели требуется, чтобы были определены только три параметра:

- ω интенсивность летальных ударов;
- ρ интенсивность нелетальных ударов;
- γ условная вероятность отказа указанного компонента, получившего нелетальный удар.

На рисунке D.2 приведен пример реализации данного метода при использовании дерева отказов.

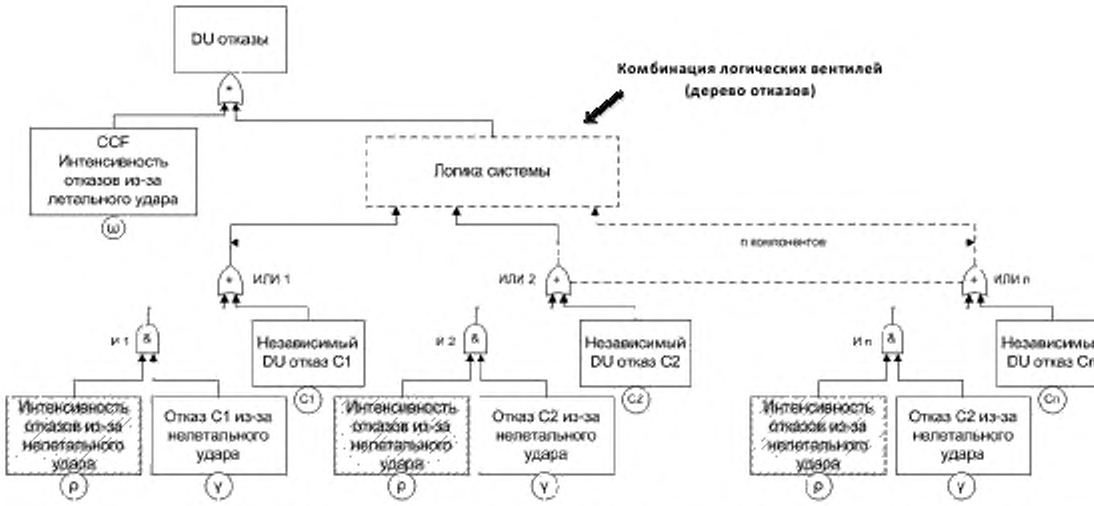


Рисунок D.2 – Реализация шоковой модели при использовании дерева отказов

Идентичные компоненты могут быть связаны с моделью β путем разделения β на две части β_L и β_{NL}:

- $\beta = \beta_L + \beta_{NL}$;
- интенсивность отказов из-за летального удара: $\lambda_{DU} \times \beta_L$;
- интенсивность отказов из-за нелетального удара: $\lambda_{DU} \times \beta_{NL}$;
- независимая интенсивность отказов: $\lambda_{DU} [1 - (\beta_L + \beta_{NL})]$.

В дереве отказов, представленном на рисунке D.2, это соответствует:

- интенсивности летальных ударов: $\omega = \lambda_{DU} \times \beta_L$;
- интенсивности нелетальных ударов: $\rho = \lambda_{DU} \times \beta_{NL} / \gamma$.

Обычно основной сложностью является вычисление значений трех параметров (ω, ρ, γ) или (β_L , β_{NL}, $\overline{\gamma}$). В [6] приводятся некоторые показатели и предоставлены другие источники информации о статистической обработке данных, позволяющей вычислить (ω, ρ, γ) из испытаний реальной системы.

Если данные отсутствуют, то возможно использование инженерских суждений при прагматическом подходе. Например, может применяться следующая процедура при использовании дерева отказов, когда в наличии больше три похожих элементов:

1 Рассматривать β, как в методе β-фактора.

2 Считать β_L незначительным (β_{NL}=β).

3 Оценить γ, чтобы была уверенность, что получен консервативный результат. Считая, что двойные отказы реализуются, по крайней мере, в 10 раз чаще, чем

четырехкратные (безусловно, консервативная гипотеза), можно использовать следующую формулу

$$\gamma = \sqrt{\frac{C_N^2}{10.C_N^4}},$$

где N – количество похожих элементов;

C_N^2 – количество потенциальных двойных отказов;

C_N^4 – количество потенциальных четырехкратных отказов.

4 Рассчитать ρ как функцию от количества N похожих элементов

$$\rho = \frac{\beta\lambda_{DU}}{C_N^2\gamma^2 + C_N^3\gamma^3}.$$

В этом методе основной вклад вносят двойные и тройные отказы, а результаты являются консервативными по сравнению с результатами, полученными при помощи метода β-фактора только с тремя компонентами. Двойные и тройные CCF рассматриваются корректно, но и маловероятные множественные отказы не полностью игнорируются.

Данная модель может быть очень легко реализована при вычислениях для моделей дерева отказов, подобно тем, что показаны в приложении В, например, для дерева отказов в В.4.3. Она позволяет очень легко анализировать системы безопасности, содержащие много похожих компонентов.

D.8 Ссылки

Полезная информация, связанная с отказами по общей причине, содержится в [17] – [21].

Приложение Е
(справочное)

Применение таблиц полноты безопасности программного обеспечения в соответствии с МЭК 61508-3

Е.1 Общие положения

Настоящее приложение содержит два примера применения таблиц полноты безопасности программного обеспечения, определенных в МЭК 61508-3, приложение А:

a) Уровень полноты безопасности 2: программируемая электронная система, связанная с безопасностью, которая используется для управления процессом на химическом заводе;

b) Уровень полноты безопасности 3: программное приложение, разработанное на языке программирования высокого уровня, которое управляет закрывающим устройством.

Данные примеры показывают, как можно выбрать методики разработки программного обеспечения в определенных обстоятельствах из таблиц приложений А и В стандарта МЭК 61508-3.

Следует подчеркнуть, что эти иллюстрации не являются безусловным применением стандартов в данных примерах. В МЭК 61508-3 в нескольких местах четко сказано, что с учетом огромного количества факторов, которые могут повлиять на системные возможности программного обеспечения, невозможно предоставить алгоритм для объединения методов и мер, которые необходимо применять для любого применения.

Все исходные характеристики конкретной системы, необходимые для использования упомянутых выше таблиц полноты безопасности, должны иметь документальное обоснование, подтверждающее, что все описания используемых характеристик правильны и соответствуют конкретной реализации этой системы. Желательно, чтобы эти обоснования опирались на ссылки на руководство в МЭК 61508-3, приложение С, в котором обсуждаются желаемые свойства, которые, если достигнуты на соответствующей стадии жизненного цикла, могут убедительно обосновать уверенность, что созданное программное обеспечение обладает достаточной систематической полнотой безопасности.

Е.2 Система с уровнем полноты безопасности 2

Пример представляет собой программируемую электронную систему, связанную с безопасностью, с уровнем полноты безопасности 2, которая используется для

управления процессом на химическом заводе. Данная программируемая электронная система использует в прикладной программе язык многозвенных логических схем и служит примером прикладного программирования на языке с ограниченной изменчивостью.

Установка, работающая на химическом заводе, состоит из нескольких реакторных баков, связанных промежуточными баками хранения, которые на некоторых стадиях цикла реакции заполняются инертным газом для предотвращения воспламенения и взрывов. Функции программируемой электронной системы, связанной с безопасностью, помимо прочих, включают в себя: получение входных данных от датчиков; включение и блокировку клапанов, насосов и исполнительных механизмов; обнаружение опасных ситуаций и включение сигнала тревоги; сопряжение с распределенной системой управления в соответствии с требованиями, предъявляемыми спецификацией безопасности.

Предположения и характеристики системы:

- программируемая электроника системы, связанной с безопасностью, представляет собой программируемый логический контроллер (ПЛК);

- при анализе опасностей и рисков установлено, что необходимо использовать программируемую электронную систему, связанную с безопасностью, и для данного приложения нужен уровень полноты безопасности 2 (в соответствии с МЭК 61508-1 и МЭК 61508-2);

- хотя контроллер работает в реальном времени, требуется относительно небольшая скорость реакции;

- существуют интерфейсы с оператором и распределенной системой управления;

- исходный код программного обеспечения системы и схема программируемых электронных средств ПЛК недоступны для проверки, но оценены в соответствии с МЭК 61508 -3 как соответствующие уровню полноты безопасности 2;

- в качестве языка программирования применения использовался язык многозвенных логических схем; программа создавалась с помощью системы разработки, предоставляемой поставщиком ПЛК;

- код приложения должен исполняться только на ПЛК одного типа;

- вся разработка программного обеспечения контролировалась лицом, независимым от команды разработчиков программного обеспечения;

- лицо, независимое от команды разработчиков программного обеспечения, наблюдало за приемочными испытаниями и утвердило их результаты;

- изменения (если необходимы) санкционируются лицом, независимым от команды разработчиков программного обеспечения.

Примечания

1 Определение независимого лица - в соответствии с МЭК 61508-4.

2 См. примечания к 7.4.2, 7.4.3, 7.4.4 и 7.4.5 в МЭК 61508-3 для информации о разделении ответственности между поставщиком ПЛК и пользователями при использовании языков программирования с ограниченной изменчивостью.

Интерпретация МЭК 61508-3, приложение А, для данного примера представлена в следующих таблицах.

Таблица Е.1 – Спецификация требований к безопасности программного обеспечения (см. МЭК 61508-3, подраздел 7.2)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1a Полуформальные методы	Таблица B.7	R	Обычно используются причинно-следственные диаграммы, циклограммы и функциональные блоки, используемые для спецификации требований к программному обеспечению ПЛК
1b Формальные методы	B.2.2, C.2.4	R	Не используются для языков программирования с ограниченной изменчивостью
2 Прямая прослеживаемость между требованиями к системе безопасности и требованиями к программному обеспечению системы безопасности	C.2.11	R	Проверка полноты: проверка, гарантирующая, что все требования к системе безопасности учтены в требованиях к программному обеспечению системы безопасности
3 Обратная прослеживаемость между требованиями к системе безопасности и предполагаемыми потребностями безопасности	C.2.11	R	Минимизация сложности и функциональности: проверка, гарантирующая, что все требования к программному обеспечению системы безопасности фактически необходимы, чтобы учесть требования к системе безопасности
4 Автоматизированные средства разработки спецификаций для поддержки, перечисленных выше, подходящих методов/средств	B.2.4	R	Используются средства разработки, поставленные производителем ПЛК

Примечания

1 В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.

2 Требования к безопасности программного обеспечения определены на естественном языке.

Таблица Е.2 – Программное обеспечение, проектирование и разработка: архитектура (см. МЭК 61508-3, пункт 7.4.3)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1 Обнаружение и диагностика сбоев	С.3.1	R	Проверка диапазона данных, контрольный таймер, ввод/вывод, средства связи. В случае ошибки поднимает тревогу (см. 3a)
2 Коды обнаружения и исправления ошибок	С.3.2	R	Встраивается с пользовательскими функциями: требуется тщательный выбор
3a Программирование с проверкой ошибок	С.3.3	R	Выделяет часть многозвенной логической схемы ПЛК для проверки некоторых важных условий безопасности (см. 1)
3b Методы контроля (при реализации процесса контроля и контролируемой функции на одном компьютере обеспечивается их независимость)	С.3.4	R	Не предпочитается: обеспечение гарантии независимости приводит к увеличению сложности программного обеспечения
3c Методы контроля (реализация процесса контроля и контролируемой функции на разных компьютерах)	С.3.4	R	Проверяет разрешенные комбинации ввода/вывода на мониторе независимого компьютера, обеспечивающего безопасность
3d Многовариантное программирование, реализующее одну спецификацию требований к программному обеспечению системы безопасности	С.3.5	---	Не предпочитается: недостаточное повышение безопасности по сравнению с 3c
3e Многовариантное (функциональ-но) программирование, реализующее различные спецификации требований к программному обеспечению системы безопасности	С.3.5	---	Не предпочитается: в значительной степени достигается 3c
3f Восстановление предыдущего состояния	С.3.6	R	Встраивается с пользовательскими функциями: требуется тщательный выбор
3g Проектирование программного обеспечения, не сохраняющего состояние (или проектирование ПО, сохраняющего ограниченное описание состояния)	С.2.12	---	Не используется. Управление процессом нуждается в состояниях, чтобы запоминать состояние установки
4a Механизмы повторных попыток парирования сбоя	С.3.7	R	Используется в соответствии с требованиями прикладной задачи (см. 2 и 3b)
4b Постепенное отключение функций	С.3.8	R	Не используется для программирования с ограниченной изменчивостью
5 Исправление ошибок методами искусственного интеллекта	С.3.9	NR	Не используется для программирования с ограниченной изменчивостью
6 Динамическая реконфигурация	С.3.10	NR	Не используется для программирования с ограниченной изменчивостью
Модульный подход	Табл. В.9	HR	
8 Использование доверительных/проверенных элементов программного обеспечения (если таковые имеются)	С.2.10	HR	Созданный ранее код для более ранних проектов
9 Прямая прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и архитектурой программного обеспечения	С.2.11	R	Проверка полноты: проверка, гарантирующая, что все требования к программному обеспечению системы безопасности учтены в требованиях к архитектуре программного обеспечения

Окончание таблицы Е.2

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
10 Обратная прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и архитектурой программного обеспечения	С.2.11	R	Минимизация сложности и функциональности: проверка, гарантирующая, что все требования к архитектуре программного обеспечения системы безопасности фактически необходимы, чтобы учесть требования к программному обеспечению системы безопасности
11a Структурные методы	С.2.1	HR	Методы потоков данных и логических таблиц данных могут использоваться, по крайней мере, для описания проекта архитектуры
1b Полуформальные методы	Табл. В.7	R	Могут быть использованы для интерфейса DCS
11c Формальные методы проектирования и усовершенствования	В.2.2, С.2.4	R	Редко используются для программирования с ограниченной изменчивостью
11d Автоматическая генерация программного обеспечения	С.4.6	R	Не используется для программирования с ограниченной изменчивостью
12 Автоматизированные средства разработки спецификаций и проектирования	В.2.4	R	Используются средства разработки, поставленные производителем ПЛК
13a Циклическое поведение с гарантированным максимальным временем цикла	С.3.11	HR	Не используется. Время цикла ПЛК контролируется техническими средствами
13b Архитектура с временным распределением	С.3.11	HR	Не используется. Время цикла ПЛК контролируется техническими средствами
13c Управление событиями с гарантированным максимальным временем реакции	С.3.11	HR	Не используется. Время цикла ПЛК контролируется техническими средствами
4 Статическое выделение ресурсов	С.2.6.3	R	Не используется. Вопросы о динамических ресурсах не возникают для программирования с ограниченной изменчивостью
15 Статическая синхронизация доступа к разделяемым ресурсам	С.2.6.3	---	Не используется. Вопросы о динамических ресурсах не возникают для программирования с ограниченной изменчивостью

Примечания

1 В столбце «Ссылка» "В.х.х.х", "С.х.х.х" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а " Табл. А.х", " Табл. В.х" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.

2 Требования к безопасности программного обеспечения определены на естественном языке.

Таблица Е.3 – Проектирование и разработка программного обеспечения: средства поддержки и язык программирования (см. МЭК 61508-3, пункт 7.4.4)

Метод/средство	Ссылка	УПБ 2	Интерпретация (в настоящем приложении)
1 Выбор соответствующего языка программирования	C.4.5	HR	Обычно используются многозвенные логические схемы и часто используются фирменные языки поставщика ПЛК
2 Строго типизированные языки программирования	C.4.1	HR	Не используется. Используется ПЛК – ориентированный структурированный текст [16]
3 Подмножество языка	C.4.2	---	Остерегайтесь использования сложных "макроинструкций", прерываний, которые изменяют цикл сканирования ПЛК, и т.д.
4a Сертифицированные средства и сертифицированные трансляторы	C.4.3	HR	Поставляется производителем ПЛК
4b Инструментальные средства и трансляторы: повышение уверенности на основании опыта использования	C.4.4	HR	Используются средства разработки, предлагаемые поставщиком ПЛК, а также собственные инструменты, разработанные в ходе работы над несколькими проектами
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а " Табл. A.x", " Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.3 – Проектирование и разработка программного обеспечения: подробная модель (см. МЭК 61508-3, пункты 7.4.5 и 7.4.6) (Включает проектирование систем программного обеспечения, проектирование модулей программного обеспечения и кодирование)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1a Структурные методы	C.2.1	HR	Не используется для языков программирования с ограниченной изменчивостью
b Полуформальные методы	Табл. B.7	HR	Используются причинно-следственные схемы, циклограммы, функциональные блоки, типичные для языков программирования с ограниченной изменчивостью
1c Формальные методы проектирования и усовершенствования	B.2.2, C.2.4	R	Не используется для языков программмирования с ограниченной измечивостью
2 Средства автоматизированного проектирования	B.3.5	R	Используются средства разработки, поставленные производителем ПЛК
3 Программирование с защитой	C.2.5	R	Включается в системное программное обеспечение
4 Модульный подход	Табл. B.9	HR	Используется упорядочение и группировка программы для ПЛК на многозвенных логических схемах для максимального увеличения модульности требуемых функций
5 Стандарты по проектированию и кодированию	C.2.6, Табл. B.1	HR	Используются собственные соглашения для документации и удобства эксплуатации
6 Структурное программирование	C.2.7	HR	Для рассматриваемого примера аналогично модульности
7 Использование доверительных/проверенных элементов программного обеспечения (по возможности)	C.2.10	HR	Функциональные блоки, части программ
8 Прямая прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и проектом программного обеспечения	C.2.11	R	Проверка полноты: проверка, гарантирующая, что все требования к программному обеспечению системы безопасности учтены в требованиях проектирования программного обеспечения
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а " Табл. A.x", " Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.5 – Проектирование и разработка программного обеспечения: проверка и интеграция программных модулей (см. МЭК 61508-3, пункты 7.4.7 и 7.4.8)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1 Вероятностное тестирование	С.5.1	R	Не используется для языков программирования с ограниченной изменчивостью
2 Динамический анализ и тестирование	В.6.5, Табл. В.2	HR	Используются
3 Регистрация и анализ данных	С.5.2	HR	Запись исходных данных и результатов тестирования
4 Функциональное тестирование и тестирование методом «черного ящика»	В.5.1, В.5.2, Табл. В.3	HR	Выбираются входные данные для тестирования всех заданных функциональных блоков, включая обработку ошибок. Используются: тестовые примеры, полученные с помощью причинно-следственных схем, анализ граничных значений и декомпозиция входных данных
5 Тестирование рабочих характеристик	Табл. В.6	R	Не используется для языков программирования с ограниченной изменчивостью
6 Тестирование, основанное на модели	С.5.27	R	Не используется для языков программирования с ограниченной изменчивостью
7 Тестирование интерфейса	С.5.3	R	Включено в функциональное тестирование и тестирование методом черного ящика
8 Управление тестированием и средства автоматизации	С.4.7	HR	Используются средства разработки, поставленные производителем ПЛК
9 Прямая прослеживаемость между спецификацией проекта программного обеспечения и спецификациями тестирования модуля и интеграции	С.2.11	R	Проверка полноты: проверка, гарантирующая, что запланирован соответствующий тест, чтобы исследовать функциональность всех модулей и их интеграции с соответственно связанными модулями
10 Формальная верификация	С.5.12	—	Не используется для языков программирования с ограниченной изменчивостью
Примечание - В столбце «Ссылка» "В.х.х.х", "С.х.х.х" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. А.х", "Табл. В.х" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.6 – Интеграция программируемых электронных средств (аппаратура и программное обеспечение) (см. МЭК 61508-3, подраздел 7.5)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1 Функциональное тестирование и тестирование методом черного ящика	B.5.1, B.5.2, табл. B.3	HR	Выбираются входные данные для тестирования всех заданных функциональных блоков, включая обработку ошибок. Используются: тестовые примеры, полученные с помощью причинно-следственных схем, анализ граничных значений и декомпозиция входных данных
2 Моделирование производительности	Табл. B.6	R	Если система ПЛК собирается для заводских приемочных испытаний
3 Прямая прослеживаемость между требованиями проекта системы и программного обеспечения к интеграции программных и аппаратных средств и спецификациями тестирования интеграции программных и аппаратных средств	C.2.11	R	Проверка, гарантирующая, что тесты интеграции аппаратуры и программного обеспечения являются адекватными
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.7 – Подтверждение соответствия аспектов программного обеспечения системы безопасности (см. МЭК 61508-3, подраздел 7.7)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1 Вероятностное тестирование	С.5.1	R	Не используется для языков программирования с ограниченной изменчивостью
2 Моделирование процесса	С.5.18	R	Не используется для языков программирования с ограниченной изменчивостью, но все чаще используется при разработке систем ПЛК
3 Моделирование	Табл. В.5	R	Не используется для языков программирования с ограниченной изменчивостью, но все чаще используется при разработке систем ПЛК
4 Функциональное тестирование и тестирование методом «черного ящика»	В.5.1, В.5.2, табл. В.3	HR	Выбираются входные данные для тестирования всех заданных функциональных блоков, включая обработку ошибок. Используются: тестовые примеры, полученные с помощью причинно-следственных схем, анализ граничных значений и декомпозиция входных данных
5 Прямая прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и планом подтверждения соответствия программного обеспечения системы безопасности	С.2.11	R	Проверка полноты: проверка, гарантирующая, что запланированное адекватное подтверждение соответствия программного обеспечения учтено в требованиях к программному обеспечению системы безопасности
6 Обратная прослеживаемость между планом подтверждения соответствия программного обеспечения системы безопасности и спецификацией требований к программному обеспечению системы безопасности	С.2.11	R	Минимизация сложности: проверка, гарантирующая, что все проверки подтверждения соответствия необходимы
Примечание - В столбце «Ссылка» "В.х.х.х", "С.х.х.х" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. А.х", "Табл. В.х" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.8 – Модификация программного обеспечения (см. МЭК 61508-3, подраздел 7.8)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1 Анализ влияния	С.5.23	HR	Выполняют анализ последствий для изучения того, насколько влияние предлагаемых изменений ограничено модульной структурой всей системы
2 Повторная верификация измененных программных модулей	С.5.23	HR	Повторение предыдущих тестов
3 Повторная верификация программных модулей, на которые оказывают влияние изменения в других модулях	С.5.23	HR	Повторение предыдущих тестов
4a Повторное подтверждение соответствия системы в целом	Табл. А.7	R	Если анализ последствий показал необходимость модификации системы, то после выполнения ее модификации обязательно проводится повторное подтверждение соответствия системы
4b Регрессионное подтверждение соответствия	С.5.25	HR	
5 Управление конфигурацией программного обеспечения	С.5.24	HR	Поддерживает базовую конфигурацию, изменения в ней, влияние на другие системные требования
6 Регистрация и анализ данных	С.5.2	HR	Выполняется запись исходных данных и результатов тестирования
7 Прямая прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и планом модификации программно-го обеспечения (включая повторные верификацию и подтверждение соответствия)	С.2.11	R	Соответствующие процедуры модификации, обеспечивающие достижение требований к программному обеспечению системы безопасности
8 Обратная прослеживаемость между планом модификации программного обеспечения (включая повторные верификацию и подтверждение соответствия) и спецификацией требований к программному обеспечению системы безопасности	С.2.11	R	Соответствующие процедуры модификации, обеспечивающие достижение требований к программному обеспечению системы безопасности
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. А.x", "Табл. В.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.9 – Проверка программного обеспечения (см. МЭК 61508-3, п. 7.9)

Метод/средство	Ссылка	УПБ	Интерпретация (в настоящем
1 Формальное доказательство	C.5.12	R	Не используется для языков программирования с ограниченной изменчивостью
2 Анимация спецификации и тестирования	C.5.26	R	
3 Статический анализ	B.6.4 Табл. B.8	HR	Выполняют анализ перекрестных ссылок использования переменных, условий и т.д.
4 Динамический анализ и тестирование	B.6.5 Табл. B.2	HR	Используются автоматические средства тестирования для облегчения регрессивного тестирования
5 Прямая прослеживаемость между спе-цификацией проекта программного обеспечения и планом верификации (включая верификацию данных) программного обеспечения	C.2.11	R	Проверка полноты: проверка, гарантирующая соответствующий тест функциональности
6 Обратная прослеживаемость между планом верификации (включая верификацию данных) программного обеспечения и спецификацией проекта программного обеспечения	C.2.11	R	Минимизация сложности: проверка, гарантирующая, что все тесты проверки необходимы
7 Численный анализ в автономном режиме	C.2.13	R	Не используется. Числовая устойчивость вычислений в данном случае не является главной проблемой
Тестирование и интеграция программных модулей	См. таблицу Е.5 настоящего стандарта		
Тестирование интеграции программируемой электроники	См. таблицу Е.6 настоящего стандарта		
Тестирование программной системы (подтверждение соответствия)	См. таблицу Е.7 настоящего стандарта		
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.10 – Оценка функциональной безопасности (см. МЭК 61508-3, раздел 8)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1 Таблица контрольных проверок	B.2.5	R	Используется
2 Таблицы решений и таблицы истинности	C.6.1	R	Используется ограниченно
3 Анализ отказов	Таблица B.4	R	На системном уровне анализ отказов использует причинно-следственные схемы, но для языков программирования с ограниченной изменчивостью этот метод не используется
4 Анализ отказов по общей причине для различного программного обеспечения (если различное программное обеспечение используется)	C.6.3	R	Не используется для языков программирования с ограниченной изменчивостью
5 Структурные схемы надежности	C.6.4	R	Не используется для языков программирования с ограниченной изменчивостью
6 Прямая прослеживаемость между требованиями раздела 8 и планом оценки функциональной безопасности программного обеспечения	C.2.11	R	Проверка полноты охвата оценки функциональной безопасности
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Е.3 Система с уровнем полноты безопасности 3

Второй пример представляет собой программное приложение уровня полноты безопасности 3, разработанное на языке программирования высокого уровня, которое управляет закрывающим устройством.

Рассматриваемая программная система сравнительно велика с точки зрения системы безопасности, так как включает более 30000 строк исходного кода. Кроме того, в ней используются обычные встроенные функции, по крайней мере, две различные операционные системы и уже существующий код более ранних проектов (проверенных в эксплуатации). В целом система состоит более чем из 100000 строк исходного кода.

Аппаратные средства (включая датчики и исполнительные механизмы) представляют собой двухканальную систему, выходы которой подключены к исполнительным элементам по схеме логического "И" (AND).

Предположения и характеристики системы:

- немедленная реакция не требуется, но обеспечивается максимальное время реакции;

- интерфейсы с оператором существуют для датчиков, исполнительных механизмов и оповещателей;

- исходный код операционных систем, графических процедур и коммерческих программных продуктов недоступен;

- система, скорее всего, в дальнейшем будет модернизироваться;

- специально разработанное программное обеспечение использует один из распространенных процедурных языков;

- компоненты программной системы, исходный код для которых недоступен, реализованы разными способами с помощью инструментальных средств от разных поставщиков, и их объектный код был создан разными трансляторами;

- программное обеспечение работает на нескольких процессорах, доступных на рынке в соответствии с требованиями МЭК 61508-2;

- встроенные системы соответствуют требованиям МЭК 61508-2 для управления отказами аппаратных средств и для их предотвращения;

- разработка программного обеспечения контролировалась независимой организацией.

Примечание - Определение независимой организации приведено в МЭК 61508-4, п. 3.8.12.

Интерпретация МЭК 61508-3, приложение А, для данного примера представлена в следующих таблицах.

Таблица Е.11 – Спецификация требований к безопасности программного обеспечения (см. МЭК 61508-3, подраздел 7.2)

Метод/средство	Ссылка	УПБ2	Интерпретация (в настоящем приложении)
1a Полуформальные методы	Табл. В.7	HR	Диаграммы функциональных блоков, циклограммы, диаграммы переходов
1b Формальные методы	В.2.2, С.2.4	R	Лишь в исключительных случаях
2 Прямая прослеживаемость между требованиями к системе безопасности и требованиями к программному обеспечению системы безопасности	С.2.11	HR	Проверка полноты: проверка, гарантирующая, что все требования к системе безопасности учтены в требованиях к программному обеспечению системы безопасности
3 Обратная прослеживаемость между требованиями к системе безопасности и предполагаемыми потребностями в безопасности	С.2.11	HR	Минимизация сложности и функциональности: проверка, гарантирующая, что все требования к программному обеспечению системы безопасности фактически необходимы, чтобы учесть требования к системе безопасности
4 Автоматизированные средства разработки спецификаций для поддержки, перечисленных выше, подходящих методов/средств	В.2.4	HR	Средства поддержки выбранных методов
Примечание - В столбце «Ссылка» "В.х.х.х", "С.х.х.х" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. А.х", "Табл. В.х" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.12 – Проектирование и разработка программного обеспечения: проектирование архитектуры программного обеспечения (см. МЭК 61508-3, пункт 7.4.3)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Обнаружение и диагностика сбоев	С.3.1	HR	Используется для тех отказов датчиков, исполнительных механизмов и средств передачи данных, которые не охватываются средствами встроенной системы в соответствии с МЭК 61508-2
2 Коды обнаружения и исправления ошибок	С.3.2	R	Используется только для внешней передачи данных
3a Программирование с проверкой ошибок	С.3.3	R	Используется для проверки подтверждения соответствия результатов прикладных функций
3b Методы контроля (при реализации процесса контроля и контролируемой функции на одном компьютере обеспечивается их независимость)	С.3.4	R	Не предпочитается: обеспечение гарантии независимости приводит к увеличению сложности программного обеспечения
3c Методы контроля (реализация процесса контроля и контролируемой функции на разных компьютерах)	С.3.4	R	Используются для некоторых функций, связанных с безопасностью, где 3a не применимы
3d Многовариантное программирование, реализующее одну спецификацию требований к программному обеспечению системы безопасности	С.3.5	---	Используются для некоторых функций, когда исходные коды не доступны
3e Многовариантное (функционально) программирование, реализующее различные спецификации требований к программному обеспечению системы безопасности	С.3.5	R	Не предпочитается: в значительной степени достигается 3c
3f Восстановление предыдущего состояния	С.3.6	---	Не используется
3g Проектирование программного обеспечения, не сохраняющего состояние (или проектирование ПО, сохраняющего ограниченное описание состояния)	С.2.12	R	Не используется. Управление закрытием нуждается в состояниях, чтобы запоминать состояние установки
4a Механизмы повторных попыток парирования сбоя	С.3.7	---	Не используется
4b Постепенное отключение функций	С.3.8	HR	Используется вследствие природы технического процесса
5 Исправление ошибок методами искусственного интеллекта	С.3.9	NR	Не используется
6 Динамическая реконфигурация	С.3.10	NR	Не используется
Модульный подход	Табл. В.9	HR	Необходимо использовать вследствие размера системы
8 Использование доверительных/проверенных элементов программного обеспечения (если таковые имеются)	С.2.10	HR	Существующий ранее код из более ранних проектов
9 Прямая прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и архитектурой программного обеспечения	С.2.11	HR	Проверка полноты: проверка, гарантирующая, что все требования к программному обеспечению системы безопасности учтены в требованиях к архитектуре программного обеспечения системы безопасности
10 Обратная прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и архитектурой программного обеспечения	С.2.11	HR	Минимизация сложности и функциональности: проверка, гарантирующая, что все требования к архитектуре программного обеспечения системы безопасности фактически необходимы, чтобы учесть требования к программному обеспечению системы безопасности

Окончание таблицы Е.12

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
11a Структурные методы	С.2.1	HR	Необходимо использовать вследствие размера системы
1b Полуформальные методы	Табл. В.7	HR	Диаграммы функциональных блоков, циклограммы, диаграммы переходов
11c Формальные методы проектирования и усовершенствования	В.2.2, С.2.4	R	Не используется
11d Автоматическая генерация программного обеспечения	С.4.6	R	Не используется. Избегайте неопределенности транслятор / генератор
12 Автоматизированные средства разработки спецификаций и проектирования	В.2.4	HR	Средства поддержки выбранных методов
13a Циклическое поведение с гарантированным максимальным временем цикла	С.3.11	HR	Не используется
13b Архитектура с временным распределением	С.3.11	HR	Не используется
13c Управление событиями с гарантированным максимальным временем реакции	С.3.11	HR	Не используется
4 Статическое выделение ресурсов	С.2.6.3	HR	Не используется. Выберите язык программирования, чтобы избежать проблемы динамических ресурсов
15 Статическая синхронизация доступа к разделяемым ресурсам	С.2.6.3	R	Не используется. Выберите язык программирования, чтобы избежать проблемы динамических ресурсов
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.13 – Проектирование и разработка программного обеспечения: средства поддержки и язык программирования (см. МЭК 61508-3, пункт 7.4.4)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Выбор соответствующего языка программирования	С.4.5	HR	Выбрается язык высокого уровня с полной изменчивостью
2 Строго типизированные языки программирования	С.4.1	HR	Используют
3 Подмножество языка	С.4.2	HR	Определяют подмножество выбранного языка
4a Сертифицированные средства и сертифицированные трансляторы	С.4.3	HR	Недоступны
4b Инструментальные средства и трансляторы: повышение уверенности на основании опыта использования	С.4.4	HR	Доступны и используют
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.14 – Проектирование и разработка программного обеспечения: подробная модель (см. МЭК 61508-3, пункты 7.4.5 и 7.4.6) (Включает проектирование систем программного обеспечения, проектирование модулей программного обеспечения и кодирование)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1a Структурные методы	C.2.1	HR	Широко используются. В частности, SADT и JSD
b Полуформальные методы	Табл. B.7	HR	Используются конечные автоматы/диаграммы перехода состояний, блок-схемы, циклограммы
1c Формальные методы проектирования и усовершенствования	B.2.2, C.2.4	R	Используются только в исключительных случаях для некоторых очень важных компонентов
2 Средства автоматизированного проектирования	B.3.5	HR	Используются для выбранных методов
3 Программирование с защитой	C.2.5	HR	В прикладном программном обеспечении в явном виде используются средства, которые могут быть эффективны, кроме автоматически вставляемых компилятором
4 Модульный подход	Табл. B.9	HR	Используются: ограниченный размер программного модуля, скрытие информации / инкапсуляция, одна входная / выходная точка в подпрограммах и функциях, полностью определенный интерфейс и т. д.
5 Стандарты по проектированию и кодированию	C.2.6, Табл. B.1	HR	Используются стандарты (предприятия) для кодирования; ограниченно используются прерывания, указатели и рекурсии; не используются динамические объекты и переменные, безусловные переходы и т. д.
6 Структурное программирование	C.2.7	HR	Используют
7 Использование доверительных/проверенных элементов программного обеспечения (по возможности)	C.2.10	HR	Доступен и используют
10 Обратная прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и архитектурой программного обеспечения	C.2.11	HR	Минимизация сложности и функциональности: проверка, гарантирующая, что все треб-вания к архитектуре программного обеспечения системы безопасности фактически необходимы, чтобы учесть требования к программному обеспечению системы безопасности
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях B и C МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях A и B МЭК 61508-3.			

Таблица Е.15 – Проектирование и разработка программного обеспечения: проверка и интеграция программных модулей (см. МЭК 61508-3, пункты 7.4.7 и 7.4.8)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Вероятностное тестирование	C.5.1	R	Используется для программных модулей, исходный код которых недоступен, а определение граничных значений и классов эквивалентности для тестовых данных затруднено
2 Динамический анализ и тестирование	B.6.5, Табл. B.2	HR	Используются для программных модулей, исходный код которых доступен. Выполняют: контрольные примеры, разработанные с помощью анализа граничных значений, моделирование производительности, разделение входных данных на классы эквивалентности, структурное тестирование
3 Регистрация и анализ данных	C.5.2	HR	Используют запись входных данных и результатов тестирования
4 Функциональное тестирование и тестирование методом «черного ящика»	B.5.1, B.5.2, Табл. B.3	HR	Используют для программных модулей, исходный код которых не доступен, и для проверки интеграции. Выбираются входные данные для тестирования всех заданных функциональных блоков, включая обработку ошибок. Используются: тестовые примеры, полученные с помощью причинно-следственных схем, прототипирование, анализ граничных значений, разделение данных на классы эквивалентности и декомпозиция входных данных
5 Тестирование производительности	Табл. B.6	HR	Используют при проверке интеграции на конкретном оборудовании
6 Тестирование, основанное на модели	C.5.27	HR	Не используют
7 Тестирование интерфейса	C.5.3	HR	Включено в функциональное тестирование и тестирование методом «черного ящика»
8 Управление тестированием и средства автоматизации	C.4.7	HR	Используются, где доступно
9 Прямая прослеживаемость между спецификацией проекта программного обеспечения и спецификациями тестирования модуля и интеграции	C.2.11	HR	Проверка, гарантирующая, что тесты интеграции достаточны
10 Формальная верификация	C.5.12	R	Не используется
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.16 – Интеграция программируемых электронных средств (аппаратура и программное обеспечение) (см. МЭК 61508-3, подраздел 7.5)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Функциональное тестирование и тестирование методом черного ящика	B.5.1, B.5.2, табл. B.3	HR	Используют как дополнительные тесты при интеграции программного обеспечения (см. таблицу Е.15). Выбираются входные данные для тестирования всех заданных функциональных блоков, включая обработку ошибок. Используются: тестовые примеры, полученные с помощью причинно-следственных схем, прототипирование, анализ граничных значений, разделение данных на классы эквивалентности и декомпозиция входных данных
2 Моделирование производительности	Табл. B.6	HR	Широко используют
3 Прямая прослеживаемость между требованиями проекта системы и программного обеспечения к интеграции программных и аппаратных средств и спецификациями тестирования интеграции программных и аппаратных средств	C.2.11	HR	Проверка, гарантирующая, что тесты интеграции аппаратуры и программного обеспечения являются достаточными
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.17 – Подтверждение соответствия аспектов программного обеспечения системы безопасности (см. МЭК 61508-3, подраздел 7.7)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Вероятностное тестирование	С.5.1	R	Не используют для подтверждения соответствия
2 Моделирование процесса	С.5.18	HR	Конечные автоматы, моделирование производительности, прототипирование и анимация
3 Моделирование	Табл. В.5	HR	Не используют для подтверждения соответствия
4 Функциональное тестирование и тестирование методом «черного ящика»	В.5.1, В.5.2, табл. В.3	HR	Выбираются входные данные для тестирования всех заданных функциональных блоков, включая обработку ошибок. Используются: тестовые примеры, полученные с помощью причинно-следственных схем, анализ граничных значений и декомпозиция входных данных
5 Прямая прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и планом подтверждения соответствия программного обеспечения системы безопасности	С.2.11	HR	Проверка полноты: проверка, гарантирующая, что все требования к программному обеспечению системы безопасности учтены в плане подтверждения соответствия программного обеспечения системы безопасности
6 Обратная прослеживаемость между планом подтверждения соответствия программного обеспечения системы безопасности и спецификацией требований к программному обеспечению системы безопасности	С.2.11	HR	Минимизация сложности: проверка, гарантирующая, что все тесты подтверждения соответствия необходимы
Примечание - В столбце «Ссылка» "В.х.х.х", "С.х.х.х" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. А.х", "Табл. В.х" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.18 — Модификация программного обеспечения (см. МЭК 61508-3, подраздел 7.8)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Анализ влияния	С.5.23	HR	Используют
2 Повторная верификация измененных программных модулей	С.5.23	HR	Используют
3 Повторная верификация программных модулей, на которые оказывают влияние изменения в других модулях	С.5.23	HR	Используют
4a Повторное подтверждение соответствия системы в целом	Табл. А.7	HR	Использование зависит от результатов анализа последствий
4b Регрессионное подтверждение соответствия	С.5.25	HR	Используют
5 Управление конфигурацией программного обеспечения	С.5.24	HR	Используют
6 Регистрация и анализ данных	С.5.2	HR	Используют
7 Прямая прослеживаемость между спецификацией требований к программному обеспечению системы безопасности и планом модификации программного обеспечения (включая повторные верификацию и подтверждение соответствия)	С.2.11	HR	Проверка полноты: проверка, гарантирующая, что процедуры модификации обеспечивают достижение требований к программному обеспечению системы безопасности
8 Обратная прослеживаемость между планом модификации программного обеспечения (включая повторные верификацию и подтверждение соответствия) и спецификацией требований к программному обеспечению системы безопасности	С.2.11	HR	Минимизация сложности: проверка, гарантирующая, что все процедуры модификации необходимы
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. А.x", "Табл. В.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.19 – Верификация программного обеспечения (см. МЭК 61508-3, п. 7.9)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Формальное доказательство	C.5.12	R	Используется только в исключительных случаях для некоторых очень важных классов
2 Анимация спецификации и тестирования	C.5.26	R	Не используется
3 Статический анализ	B.6.4 Табл. B.8	HR	Для всего вновь разработанного кода используются: анализ граничных значений, таблица контрольных проверок, анализ потоков управления, анализ потоков данных, проверка разработки программ, анализ проектов
4 Динамический анализ и тестирование	B.6.5 Табл. B.2	HR	Для всего вновь разработанного кода
5 Прямая прослеживаемость между спецификацией проекта программного обеспечения и планом верификации (включая верификацию данных) программного обеспечения	C.2.11	HR	Проверка полноты: проверка, гарантирующая, что процедуры модификации обеспечивают достижение требований к программному обеспечению системы безопасности
6 Обратная прослеживаемость между планом верификации (включая верификацию данных) программного обеспечения и спецификацией проекта программного обеспечения	C.2.11	HR	Минимизация сложности: проверка, гарантирующая, что все процедуры модификации необходимы
7 Численный анализ в автономном режиме	C.2.13	HR	Не используется. Числовая устойчивость вычислений в данном случае не является главной проблемой
Тестирование и интеграция программных модулей	См. таблицу Е.15 настоящего стандарта		
Тестирование интеграции программируемой электроники	См. таблицу Е.16 настоящего стандарта		
Тестирование программной системы (подтверждение соответствия)	См. таблицу Е.17 настоящего стандарта		
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Таблица Е.20 – Оценка функциональной безопасности (см. МЭК 61508-3, раздел 8)

Метод/средство	Ссылка	УПБЗ	Интерпретация (в настоящем приложении)
1 Таблица контрольных проверок	B.2.5	R	Используют
2 Таблицы решений и таблицы истинности	C.6.1	R	Используют в ограниченной степени
3 Анализ отказов	Табл. B.4	HR	Интенсивно используют анализ диагностического дерева отказов, а причинно-следственные диаграммы используют в ограниченной степени
4 Анализ отказов по общей причине для различного программного обеспечения (если различное программное обеспечение используется)	C.6.3	HR	Используют
5 Структурные схемы надежности	C.6.4	R	Используют
6 Прямая прослеживаемость между требованиями раздела 8 и планом оценки функциональной безопасности программного обеспечения	C.2.11	HR	Проверка полноты охвата оценки функциональной безопасности
Примечание - В столбце «Ссылка» "B.x.x.x", "C.x.x.x" указывают на описания методов, изложенные в приложениях В и С МЭК 61508-7, а "Табл. A.x", "Табл. B.x" – на таблицы методов, представленные в приложениях А и В МЭК 61508-3.			

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК Руководство 51:1990	IDT	ГОСТ Р 51898-2002 «Аспекты безопасности. Правила включения в стандарты»
МЭК Руководство 104:1997	–	*
МЭК 61508-1:2010	IDT	ГОСТ Р МЭК 61508-1–2012 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 1. Общие требования»
МЭК 61508-2:2010	IDT	ГОСТ Р МЭК 61508-2–2012 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 2. Требования к системам»
МЭК 61508-3:2010	IDT	ГОСТ Р МЭК 61508-3–2012 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 3. Требования к программному обеспечению»
МЭК 61508-4:2010	IDT	ГОСТ Р МЭК 61508-4–2012 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 4. Термины и определения»
МЭК 61508-5:2010	IDT	ГОСТ Р МЭК 61508-5–2012 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 5. Рекомендации по применению методов определения уровней полноты безопасности»
МЭК 61508-7:2010	IDT	ГОСТ Р МЭК 61508-7–2012 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 7. Методы и средства»

*Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание – В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов:

- IDT – идентичные стандарты.

Библиография

[1] IEC 60601 (all parts), Medical electrical equipment

[2] ISA-TR84.00.02-2002 – Parts 1-5, Safety Instrumented Functions (SIF) Safety Integrity Level (SIL) Evaluation Techniques Package.

[3] IEC 61078:2006, Analysis techniques for dependability – Reliability block diagram and boolean methods

[4] IEC 61025:2006, Fault tree analysis (FTA)

[5] IEC 61165:2006, Application of Markov techniques

[6] IEC 62551, Analysis techniques for dependability – Petri Net technique

[7] BS 5760, Reliability of system equipment and components – Part 2: Guide to assessment of reliability

[8] D. J. SMITH, Reliability, maintainability and risk – Practical methods for engineers, Butterworth-Heinemann, 5th edition, 1997, ISBN 0-7506-3752-8

[9] R. BILLINGTON and R. N. ALLAN, Reliability evaluation of engineering systems, Plenum, 1992, ISBN 0-306-44063-6

[10] W. M. GOBLE, Evaluating control system reliability – Techniques and applications, Instrument Society of America, 1992, ISBN 1-55617-128-5

[11] A. ARNOLD, A. GRIFFAULT, G. POINT, AND A. RAUZY. The altarica language and its semantics. Fundamenta Informaticae, 34,pp.109–124, 2000

[12] M. BOITEAU, Y. DUTUIT, A. RAUZY AND J.-P. SIGNORET, The AltaRica Data-Flow Language in Use: Assessment of Production Availability of a MultiStates System, Reliability Engineering and System Safety, Elsevier, Vol. 91, pp 747-755

[13] A. RAUZY. Mode automata and their compilation into fault trees. Reliability Engineering and System Safety, Elsevier 2002, Volume 78, Issue 1, pp 1-12

[14] Reliability Analysis Center (RAC), Failure Mode/Mechanism Distributions , 1991, Department of Defense, United States of America, PO Box 4700, 201 Mill Street, Rome, NY 13440-8200, Organization report number: FMD-91, NSN 7540-01-280-5500

[15] ALLESSANDRO BIROLINI, Qualitдt und Zuverlassigkeit technischer Systeme, Theorie, Praxis, Management, Dritte Auflage,1991, Springer-Verlag, Berlin Heidelberg New York, ISBN 3-540-54067-9, 3 Aufl., ISBN 0-387-54067-9 3 ed. (available in German only)

[16] MIL-HDBK-217F, Military Handbook Reliability prediction of electronic equipment, 2 December 1991, Department of Defense, United States of America

[17] Health and Safety Executive Books, email hsebooks@prolog.uk.com

[18] ANIELLO AMENDOLA, kluwer academic publisher, ISPRA 16-19 November 1987, Advanced seminar on Common Cause Failure Analysis in Probabilistic Safety Assessment, ISBN 0-7923-0268-0

[19] CORWIN L. ATWOOD, The Binomial Failure Rate Common Cause Model, Technometrics May 1986 Vol 28 n°2

[20] R. HUMPHREYS, A., PROC., Assigning a numerical value to the beta factor common-cause evaluation, Reliability 1987

[21] UPM3.1, A pragmatic approach to dependent failures assessment for standard systems, AEA Technology, Report SRDA-R-13, ISBN 085 356 4337, 1996

[22] For PDS method; see (www.sintef.no/pds); and further background material in:Hokstad, Per; Corneliussen, Kjell Source: Reliability Engineering and System Safety, v 83, n 1, p. 111-120, January 2004

УДК 62-783:614.8:331.454:006.354 ОКС 13.110 Группа Т51

Ключевые слова: функциональная безопасность; жизненный цикл систем; электрические компоненты; электронные компоненты; программируемые электронные компоненты и системы; системы, связанные с безопасностью; охват диагностикой; оценка вероятности отказа аппаратных средств; полнота безопасности программного обеспечения

Подписано в печать 30.04.2014. Формат 60x84 $^{1}/_{8}$.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ФГУП «СТАНДАРТИНФОРМ»

123995 Москва, Гранатный пер., 4.

www.gostinfo.ru info@gostinfo.ru