

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

**ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
РАДИОЭЛЕКТРОНИКИ**

ХАХАНОВ В.И.

ЛИТВИНОВА Е.И.

ЧУМАЧЕНКО С.В.

Надежность и тестопригодность микросистем

Microsystems Reliability and Testability

**Рекомендовано к опубликованию Ученым Советом
Харьковского Национального Университета
Радиоэлектроники**

№530785-TEMPUS-1-2012-1-PL-TEMPUS-JPCR



Tempus

Харьков 2016

Навчальний посібник **"Надійність і тестопридатність мікросистем"** створено для допомоги вищим навчальним закладам України впровадити нову магістерську навчальну програму "Проектування мікросистем".

Посібник **"Надійність і тестопридатність мікросистем"** створено при підтримці Європейського Союзу за Спільним Європейським Проектом "Curricula Development for New Specialization: Master of Engineering in Microsystems Design" (MastMST), ідентифікаційний номер 530785-TEMPUS-1-2012-1-PL-TEMPUS-JPCR.

Координатор проекту проф. Збігнєв Лісік, Технічний університет м.Лодзь, Польща.

Учасники проекту:

- Національний університет "Львівська політехніка", м. Львів, Україна,
- координатор проф. Михайло Лобур.
- Київський Національний університет ім. Тараса Шевченка, м. Київ, Україна,
координатор проф. Валерій Скришевський.
- Харківський національний університет радіоелектроніки, м. Харків, Україна,
координатор проф. Володимир Хаханов.
- Донецький національний технічний університет, м. Красноармійськ, Україна,
координатор проф. Володимир Святний.
- Технічний університет м. Ільменау, Німеччина,
координатор проф. Іво Рангелов.
- Ліонський Національний інститут прикладних наук, Франція,
координатор проф. Александра Апотолук
- Університет Павії, м. Павії, Італія,
координатор проф. Паоло Ді Барба

Посібник схвалено редакційним комітетом (проф. Паоло Ді Барба (Університет Павії) - співголова, проф. Александра Апотолук (Ліонський Національний інститут прикладних наук) – співголова, члени: проф. Збігнєв Лісік (Технічний університет м.Лодзь), д-р Яцек Подгурські (Технічний університет м.Лодзь), Д-р Януш Возний (Технічний університет м.Лодзь), Д-р Валентин Іщук (Технічний університет м.Ільменау), Д-р Марія-Евеліна Могначі (Університет Павії), Д-р Роберто Галді (Університет Павії)) 6 травня 2016, м. Павія, Італія

Автори висловлюють глибоку вдячність керівництву вищеназваних університетів за всебічну підтримку Проекту.

ББК 32.973.2

Б81

Надежность и тестопригодность микросистем / В.И. Хаханов, Е.И. Литвинова, С.В. Чумаченко. – Харьков: ХНУРЭ. – 2015. – 162 с.

Представлены математические основы теории надежности микросистем. Дан аналитический обзор современного состояния рынка микроэлектромеханических систем (МЭМС). Используются лучшие зарубежные и отечественные издания, государственные стандарты, а также работы ведущих ученых и специалистов в области теории надежности. Предлагаются кубитные модели и методы повышения тестопригодности программных и аппаратных средств анализа цифровых устройств за счет увеличения размерности структур данных и памяти. Введены основные понятия, термины и определения, необходимые для имплементации квантовых вычислений в практику анализа виртуальных компьютеров. Представлены результаты исследований, касающиеся моделей и методов диагностирования цифровых систем, моделирования исправного поведения, восстановления работоспособности отказавших примитивов. Описана инновационная идея квантовых вычислений, которая характеризуется переходом от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение (точку) к логическим регистровым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений, что дает возможность определить новые перспективы на пути создания высокопроизводительных компьютеров параллельного анализа и синтеза структур и сервисов дискретного киберпространства.

Книга предназначена для студентов, аспирантов и специалистов в области автоматизированного проектирования МЭМС, а также для широкого круга читателей, специализирующихся в области надежности микроэлектромеханических систем.

Рис. 52. Табл. 5. Библиогр.: 38 назв.

**В.И. Хаханов, Е.И. Литвинова,
С.В. Чумаченко, 2016**

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1. ВИДЫ НЕИСПРАВНОСТЕЙ МЭМС.....	9
1.1. Виды МЭМС	9
1.2. Компоненты МЭМС	11
1.3. Виды неисправностей электронных компонентов МЭМС	19
1.4. Модели функциональных неисправностей	22
1.5. Основные понятия, термины и определения состояний МЭМС и свойств надежности.....	26
1.4. Срок службы МЭМС	28
2. МОДЕЛИ НАДЕЖНОСТИ МЭМС.....	30
2.1. Модели надежности МЭМС	30
2.2. Распределение надежности.....	31
3. СТРУКТУРНЫЕ СХЕМЫ НАДЕЖНОСТИ МЭМС.....	38
3.1. Анализ модели надежности	38
3.2. Анализ дерева неисправностей.	41
4. РЕЗЕРВИРОВАНИЕ НЕВОССТАНАВЛИВАЕМЫХ СИСТЕМ....	53
4.1. Раздельное резервирование системы.....	53
4.2. Резервирование системы однотипных элементов	54
4.3. Дублирование.....	55
4.4. Резервирование замещением	55
4.5. Нагруженное резервирование неоднотипных элементов	56
4.6. Скользящее резервирование	56
4.7. Частные случаи	57
4.8. Автономные устройства с общим резервированием	58

4.9. Частные случаи	59
4.10. Дублирование в ненагруженном режиме	60
4.11. Произвольное число резервных устройств в ненагруженном режиме	62
4.12. Резервирование системы при экспоненциальном распределении интервалов безотказной работы ее элементов	63
4.13. Резервирование системы при неэкспоненциальном распределении интервалов безотказной работы ее элементов.....	73
4.14. Оптимизация затрат.....	76
5. НАДЕЖНОСТЬ ВОССТАНАВЛИВАЕМЫХ СИСТЕМ.....	89
5.1. Элементы теории массового обслуживания.....	89
5.2. Восстанавливаемая система без резервирования	99
5.4 Скользящее резервирование с восстановлением	110
6. КУБИТНЫЕ (КВАНТОВЫЕ) ТЕХНОЛОГИИ ПОВЫШЕНИЯ НАДЕЖНОСТИ МЭМС.....	118
6.1. Кубитные структуры данных вычислительных устройств восстановления работоспособности	118
6.2. Кубитные технологии анализа и диагностирования цифровых устройств	138
ЗАКЛЮЧЕНИЕ.....	157
8. СПИСОК ЛИТЕРАТУРЫ	160

ВВЕДЕНИЕ

Развитие микроэлектромеханических (МЭМС) технологий изменило мир и привело к появлению недорогих высоконадежных устройств со сложной функциональностью, которые широко используются в автомобильной, индустриальной, медицинской, военной отраслях, а также в потребительской электронике [1].

За последние два десятилетия микросистемы эволюционировали от миниатюрных однофункциональных устройств к гораздо более сложным интеллектуальным интегрированным системам. С точки зрения производительности различают несколько поколений интеллектуальных систем. Системы первого поколения (Smart Systems) состояли из нескольких пакетов компонентов, соединенных на одной подложке или печатной плате, и применялись в слуховых аппаратах и кардиостимуляторах, а также в автомобильных системах, например, подушках безопасности. Наиболее известным примером интеллектуальной системы второго поколения является смартфон, имеющий большой коммерческий успех.

Самодостаточные интеллектуальные технические системы третьего поколения характеризуются расширенной функциональностью, способны выполнять сбор данных, актюаторные воздействия и обработку информации. Высокая надежность и наличие адресации всех компонентов системы обеспечивают возможность идентификации отдельных устройств и их функционирование во взаимосвязи с другими устройствами. Такие интеллектуальные системы являются аппаратной основой Интернета вещей (Internet of Things, IoT) и новых приложений: микроразмерных зеркал, предназначенных для направления и/или сканирования светового луча, компактных и быстродействующих систем автофокусировки для мобильных телефонов, новых типов печатающих головок для 3D-печати.

Для функционирования технологии Internet of Things (IoT) и новой технологической культуры Smart Everything требуются миллиарды микроэлектромеханических устройств и датчиков, выполняющих функции мониторинга и управления. Структура рынка IoT представлена на рис. 1, доля его основных компонентов (hardware, cloud, data) – на рис. 2.

В настоящее время на рынке МЭМС доминируют потребительские приложения, такие как акселерометры и гироскопы, используемые в смартфонах и планшетных компьютерах, автомобильные и промышленные приложения, применяемые для реализации технологии IoT.

По прогнозам объем продаж рынка МЭМС в 2018 году достигнет 12 млрд долларов [2].

Согласно данным отчета компании Yole Developpment за 2015 год «Статус МЭМС-промышленности» в последующие шесть лет совокупный среднего-

довой темп роста сегмента пользовательских и мобильных приложений составит 11%, промышленные, медицинские, телекоммуникационные, военные, космические приложения обеспечат рост 18%, автомобильные приложения – 28%. Ожидается, что рынок МЭМС увеличится с \$10 млрд в 2015 году до \$14 млрд в 2019 году, демонстрируя в этот период совокупный ежегодный темп роста 7,6% (рис. 3) [3].

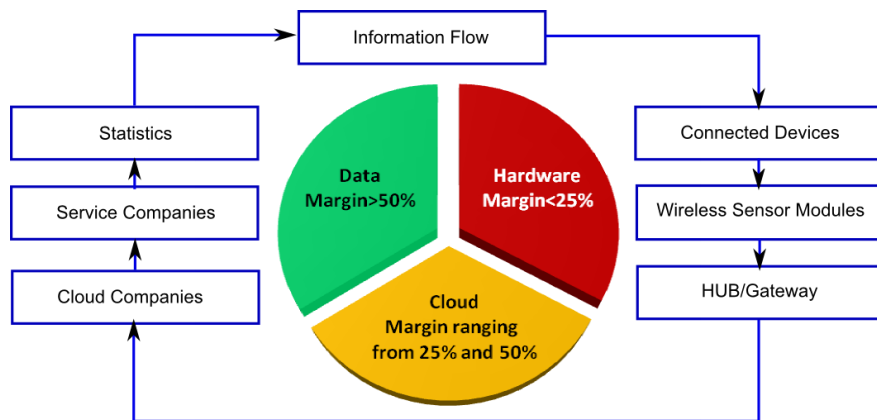


Рис. 1. Структура рынка IoT

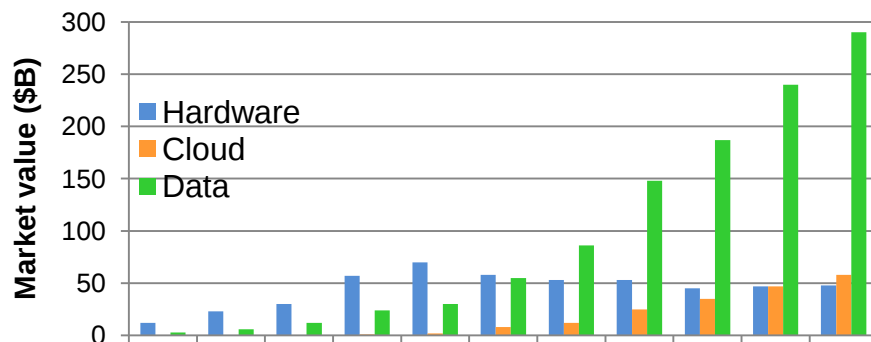


Рис. 2. Компоненты рынка IoT

Успех микроэлектромеханических систем определяется следующими критериями: малые размеры, низкая стоимость, сложная функциональность, простота производства (выход годных – Yield), низкое энергопотребление, высокая надежность.

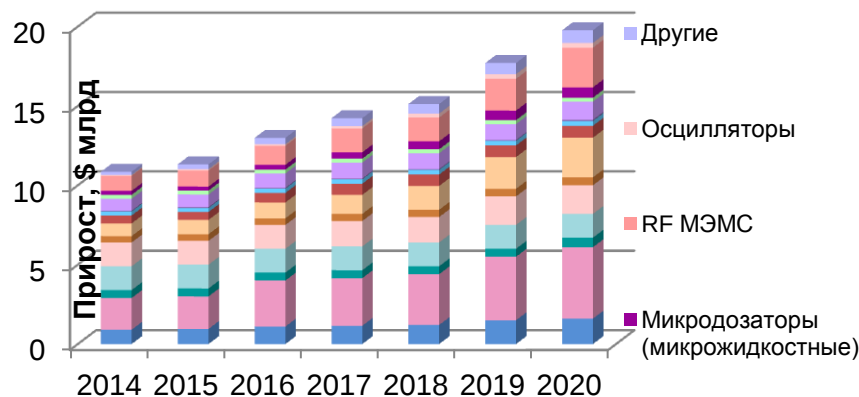


Рис. 3. Прогноз рынка МЭМС

Лидерами на рынке разработки и производства микроэлектромеханических систем (МЭМС) являются: Robert Bosch, STMicroelectronics, Texas Instruments, Hewlett Packard, Knowles Electronics.

Микроэлектромеханические системы (МЭМС) представляют собой устройства микросистемной техники, выполненные по технологии объёмной микромеханики, сформированные путём локального вытравливания подложки, легирования, нанесения на неё материала. Подложки, как правило, изготавливаются из кремния благодаря его превосходным электрическим, механическим и тепловым свойствам. Размеры МЭМС лежат в диапазоне от 1 микрометра до нескольких миллиметров, в зависимости от мощности, области применения, наличия встроенных схем обработки и количества элементов.

Основные преимущества МЭМС:

- Миниатюрность;
- Высокая функциональность;
- Надёжность;
- Малое энергопотребление;
- Возможность интеграции электроники с механическими, оптическими и прочими узлами;
- Малый разброс параметров в пределах одной партии изделий;
- Высокая технологичность и повторяемость;
- Возможность достичь очень низкую стоимость (при больших или очень больших объёмах производства).

1. ВИДЫ НЕИСПРАВНОСТЕЙ МЭМС

1.1. Виды МЭМС

МЭМС представляет собой систему высокой степени интеграции, в состав которой входят электрические, механические, магнитные, жидкостные, термические элементы, изготовленную с использованием полупроводниковых технологий, размером от нескольких нанометров до микрометров или миллиметров [4]. Микроэлектромеханические системы предназначены для взаимодействия с внешней средой в режиме мониторинга и управления в целях получения информации о состоянии объекта и формирования управляющих воздействий.

Компоненты МЭМС широко используются в различных приложениях, и условно их можно разделить на два типа: сенсоры – измерительные устройства, которые преобразуют физические воздействия в электрический сигнал, и актюаторы (исполнительные устройства), которые выполняют обратные действия – преобразуют сигналы в физические воздействия.

К сенсорам относят:

- Датчики движения (акселерометры – датчики ускорения, гироскопы – датчики поворота) – используются в мобильных телефонах, видеоприставках, портативных компьютерах, фототехнике, устройствах определения местоположения;
- Микрофоны;
- Датчики давления.

МЭМС акселерометры являются одними из наиболее простых, и одновременно наиболее часто используемых микро-электромеханических систем. Они незаменимы в автомобильной промышленности, компьютерной и аудио-видео технике [5].

Акселерометр представляет собой электромеханическое устройство, которое измеряет ускорение силы. Сила может быть статической (например, постоянная сила тяжести) или динамической (в результате перемещения или вибрации). В МЭМС акселерометрах может использоваться пьезоэлектрический эффект, когда при воздействии силы инерции на микроскопические кристаллические структуры возникает напряжение, или производится измерение изменения емкости конденсатора. Преимущество второго способа – при реализации акселерометра после выполнения основных технологических операций не требуется дополнительная обработка. Конденсаторы могут выполнять функции датчиков и актюаторов, характеризуются высокой чувствительностью и используемый в них механизм трансдукции нечувствителен к воздействию температуры.

Гироскоп представляет собой устройство для измерения угловой скорости тела относительно заданной оси вращения. Принцип работы MEMS гироскопа основан на использовании эффекта Кориолиса при вращении двух кремниевых масс, совершающих колебания равной амплитуды в противоположных направлениях. Действие силы Кориолиса вызывает ортогональную вибрацию колеблющихся масс (смещение их от плоскости в зависимости от величины угловой скорости) [6]. МЭМС гироскопы используются в автомобильной, промышленной и потребительской электронике, характеризуются низкой ценой и малыми размерами.

MEMS микрофоны используются в аудио приложениях, где основными критериями являются малый размер, высокое качество звука, надежность и доступность. MEMS микрофоны строятся на основе конденсаторного принципа или пьезоэффекта.

MEMS датчики давления широко используются в медицине, обеспечивая высокое разрешение и надежность при малых размерах.

Цифровые микрозеркальные устройства – DLP-проекторы (DLP – Digital Light Processing) – используются в цифровом кинематографе, бизнес-приложениях, образовании и домашних кинотеатрах. Преимуществами DLP-технологии являются:

- Стойкий точный цвет;
- Высокая скорость переключения цветов, обеспечиваемая встроенными интеллектуальными компонентами;
- Низкое энергопотребление;
- использование инновационной технологии управления светом DLP Cinema®.

Цифровое микрозеркальное устройство (Digital Micromirror Device, DMD) на основе технологии DLP представляет собой массив микрозеркал, функционирующий в режиме реального времени в целях эффективной и высоконадежной пространственной модуляции света. DMD устройства используются в системах цифровой литографии, машинного зрения, 3D-печати, спектроскопии.

Матрицы с количеством микрозеркал менее 1 миллиона применяются в 3D-принтерах, сканерах, спектрометрах, допускают синхронизацию и совместную работу с камерами, датчиками, шаговыми двигателями и другими внешними устройствами.

Матрицы с большим количеством микрозеркал (≥ 1 миллион) могут содержать более 2560x1600 элементов (4 миллиона пикселей), что обеспечивает высокую эффективность и яркость, а также возможность измерения больших объектов с точностью микронного уровня в 3D сканерах, анализ жидкостей и твердых веществ с высоким спектральным разрешением в спектрометрах и печать объектов с большим пятном (footprint) в 3D-принтерах.

1.2. Компоненты МЭМС

Большинство МЭМС включает следующие основные механические модули: консольные балки (с односторонним или двусторонним зажимом), мембраны (плавающие и закрытые), пружины, шарниры. Указанные элементы подвержены деградации, в них могут возникать повреждения, что приводит к отказам микросистемы в целом. В табл. 1 приведен обзор наиболее распространенных элементов, применяемых в МЭМС [7]. Анализ материалов, из которых изготовлены эти элементы, и условий окружающей среды позволяет сформировать список возможных нарушений элементов и механизмов МЭМС, приведенный в табл. 2.

Свойства материалов конструктивных элементов микросистем, изготовленных по тонкопленочной технологии, достаточно хорошо изучены. Однако проблемы деградации материалов, ползучести и усталости остаются актуальными в настоящее время. Прилипание может вызывать сбои в устройствах МЭМС, имеющих малый зазор между отдельными структурными элементами (например, при поверхностной микрообработке). Результаты исследований материалов, применяемых при изготовлении МЭМС, позволяют сделать выводы о том, какие виды отказов характерны для материалов той или иной группы.

Таблица 1 – Элементы МЭМС

Элементы МЭМС
1. Structural beams – консольные балки
- rigid (жесткий)
- flexible (гибкий)
- one side clamped (с односторонним зажимом)
- two sides clamped (с двусторонним зажимом)
2. Structural thin membranes - структурные тонкие мембраны
- rigid (жесткие)
- flexible (гибкие)
- with holes (с отверстиями)
3. Flat layers (плоские слои, приклеенные к подложке)
- conductive (проводящие)
- insulating (изоляционные)
4. Hinges (шарниры)
- Substrate hinge (шарнир на подложке)
- scissors hinge (ножничный шарнир)
5. Cavities (полости)

- sealed (запечатанная)
- open (открытая)
6. Gears (шестерни)
- teeth (зубчатые)
- hubs (ступичные)
7. Tunneling tips (туннельные зонды в виде острия)
8. Reflective layers (отражающие слои)

Таблица 2 – Механизмы неисправностей МЭМС

Механизмы неисправностей МЭМС
1. Fracture (трещина, разрушение)
- overload fracture (разрушение под воздействием чрезмерных нагрузок)
- fatigue fracture (усталостное разрушение)
2. Creep (ползучесть)
- applied stress (приложенное напряжение)
- intrinsic stress (внутреннее напряжение)
- thermal stress (термическое напряжение)
3. Stiction (прилипание)
- capillary forces (капиллярные силы)
- van der Waals molecular forces (молекулярные силы Ван-дер-Ваальса)
- electrostatic forces (электростатические силы)
- solid bridging (твердые перемычки)
4. Electromigration (электромиграция)
5. Wear (износ)
- adhesive (адгезивный)
- abrasive (абразивный)
- corrosive (коррозионный)
6. Degradation of dielectrics (деградация диэлектриков)
- leakage (утечка)
- charging (зарядка)
- breakdown (пробой)
7. Delamination (расслаивание)

8. Contamination (загрязнение)
9. Pitting of contacting surfaces (коррозия контактирующих поверхностей)
10. Electrostatic discharge, ESD (электростатический разряд)

В отличие от полупроводниковых микроэлектронных устройств, не имеющих движущихся частей, при производстве и эксплуатации МЭМС возникает несколько сложных технических проблем: 1) компоненты МЭМС могут содержать множество различных 3D-структур, зачастую очень сложной формы, с размерами порядка нескольких микрон, изготовленных из различных материалов; 2) функциональные возможности конечного продукта тесно определяются технологическим процессом его создания, основанном на принципе «один продукт – один процесс», в отличие от СБИС индустрии, где множество однотипных продукты изготавливаются с помощью одного технологического процесса. Для МЭМС не существует эквивалента 32 нм узла.

Совершенствование технологии изготовления МЭМС в последние годы позволило использовать различные материалы для создания трехмерных структур на микроуровне и стандартных процессов фотолитографии для обеспечения больших объемов производства устройств МЭМС.

Современная методология разработки МЭМС основывается на методологии проектирования сверху вниз, что позволяет уменьшить время проектирования и производства. Одной из основных задач по-прежнему является обеспечение надежности МЭМС в заданных режимах функционирования.

Традиционный алгоритм проектирования МЭМС представлен на рис. 1.1 [4]. Алгоритм проектирования МЭМС с использованием параллельных операций основан на интегрировании функций проектирования и производства в целях уменьшения времени выхода изделия на рынок time-to-market, рис. 1.2.

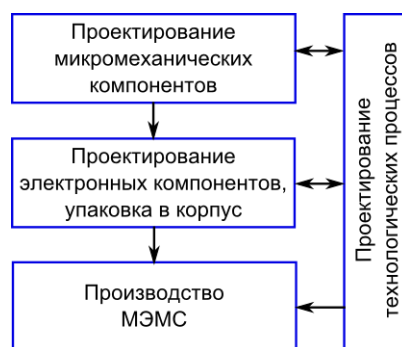


Рис. 1.1. Традиционный алгоритм проектирования МЭМС

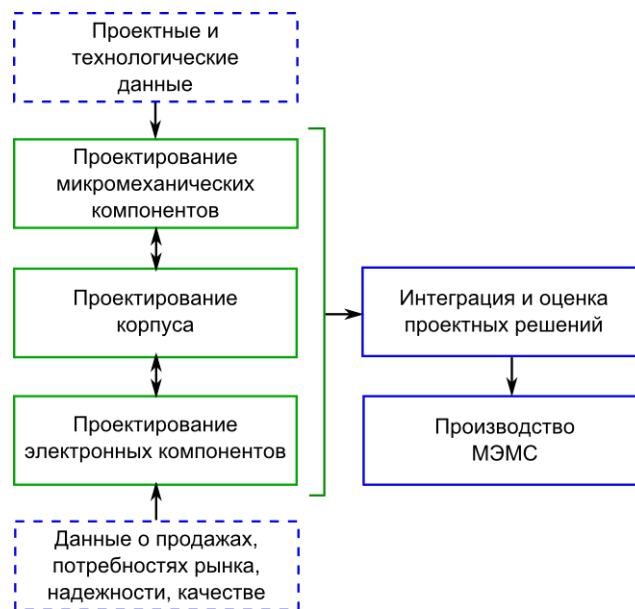


Рис. 1.2. Алгоритм проектирования МЭМС на основе параллельных операций

Проблемы проектирования и производства МЭМС [8]:

- 1) использование подхода «одно устройство – один процесс», когда для каждой МЭМС разрабатывается уникальный технологический процесс. В целях уменьшить затраты на производство и время изготовления системы необходимо использовать стандартизированные библиотечные модули технологического процесса;
- 2) статистическое управление процессами (statistical process control, SPC) зачастую невозможно. Разработчики оценивают варианты решения, основываясь на прошлом опыте;
- 3) широкое использование МЭМС в мобильных устройствах приводит к необходимости повышения уровня интеграции за счет совершенствования технологий изготовления и упаковки кристаллов в корпус (wafer bumping – формирование столбиковых выводов, die-on-wafer assembly – монтаж многокристалльных модулей, through-silicon via – формирование переходных отверстий). Для обеспечения надежности систем высокой степени интеграции необходимо использовать средства моделирования и компенсации производственных дефектов;
- 4) традиционные методы проектирования, при которых система проектируется непосредственно на низком аппаратном уровне или на уровне программного обеспечения являются неэффективными. Это приводит к разрыву между требованиями к сложности проектируемых систем и возможностями систем автоматизированного проектирования.

Одним из общепринятых решений описанных проблем является повышение уровня абстракции в процессе проектирования. Для достижения приемлемого роста производительности и преодоления семантического разрыва между высоким и низким уровнями проектирования и имплементации необходимо повышать уровень автоматизации систем проектирования. Проектировщик должен применять средства автоматизации для моделирования, симуляции, синтеза и верификации проекта. Однако внедрение средств автоматизации является не простой задачей, если уровень системных абстракций не достаточно определен, компоненты конкретного уровня абстракции не известны, языки системного уровня проектирования не имеют четкой семантики, правила проектирования и стили моделирования характеризуются высокой сложностью.

По мере развития МЭМС индустрии осуществляется переход от проектирования микроструктур к проектированию микросистемы в целом (системный подход к проектированию), который позволяет выполнять совместное проектирование и моделирование компонентов программного (software, SW) и аппаратного (hardware, HW) обеспечений. Моделирование МЭМС осуществляется на различных уровнях абстракции. Неэмпирические модели базируются на атомистической, квантово-механической или молекулярной динамике. Такие модели, как правило, используются при моделировании технологических процессов для предсказания поведения материала (например, в процессе травления). Модели компонентного уровня (с сосредоточенными параметрами и конечных элементов) позволяют описывать такие компоненты, как пластины и гребенчатый привод. Модели устройств описывают функционирование микро- или наноструктуры. Алгоритмические модели используются для описания поведения логического или управляющего элемента системы в процессе функционирования. Модели системного уровня применяются для моделирования микросистемы в целом.

X-диаграмма проектирования микро- и наносистем, иллюстрирующая уровни абстракции и аспекты проектирования представлена на рис. 1.3. На диаграмме представлены четыре аспекта проектирования: поведенческий, структурный, физический, технологический и пять уровней абстракции: системный, алгоритмический, схемный, компонентный и базовый (уровень идеи или принципа действия).

Поведенческий аспект позволяет представить проект в виде черного ящика, определить временные ограничения, ограничения по площади кристалла, потребляемой мощности, тестопригодности. Поведенческая модель обычно задается в виде исполняемых функциональных описаний на языке типа C (или C++). По этим функциональным описаниям затем выполняется моделирование для широкого набора входных воздействий. Поведенческая модель черного ящика не определяет способ реализации и структуру объекта.

Структурный аспект направлен на представление проекта в виде набора компонентов и связей между ними. Поведение черного ящика определяется

поведением его компонентов и их взаимосвязями, однако поведение проекта в целом может быть сложным для понимания, поскольку оно скрыто в деталях отдельных компонентов и связей между ними.

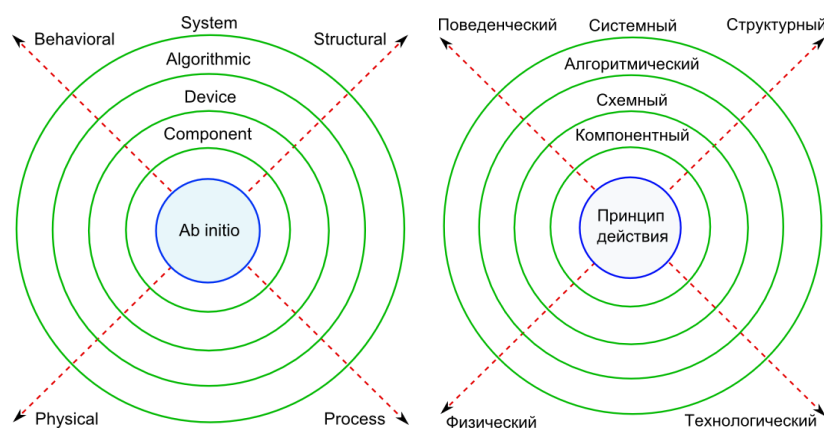


Рис. 1.3. X-диаграмма проектирования микро- и наносистем

Физический аспект определяет для каждого компонента его размеры (высоту и ширину), местоположение, порты и соединения на силиконовом кристалле или печатной плате. Физическая модель существенно увеличивает размерность структуры.

Технологический аспект направлен на моделирование процессов формирования физической геометрии микроструктуры, например, диффузии, выращивания кристалла или травления.

X-диаграмма представляет проект на различных уровнях абстракции, представленных на рис. 1.3 в виде концентрических окружностей с центром в начале координат. Название каждого уровня абстракции определяется типом компонентов, генерируемых на данном уровне.

Базовый уровень определяет устройство на основе атомистических или квантово-механических принципов молекулярной динамики. Информация базового уровня используется при прогнозировании свойств и поведения материала на наноуровне. Объекты, создаваемые на компонентном уровне, являются стандартными ячейками, состоящими из транзисторов с каналом n-типа или p-типа и базовых элементов МЭМС.

Схемотехническое проектирование с использованием компонентов реализуется на основе нисходящей методологии «сверху вниз». Одним из основных преимуществ данного подхода является то, что проект формируется из фундаментальных строительных блоков или компонентов. Это позволяет пользователю формировать параметризованные модели устройства, описывающие компоновочные и технологические данные.

Проект, представленный в виде параметризованной абстрактной модели, может быть проанализирован на разных уровнях абстракции и детализации. Для описания атомарных объектов используются модели конечных и граничных элементов FEA/BEA (Finite Element Analysis / Boundary Element Analysis) или распределенные модели; анализ объектов выполняется по методу Рэлея-Ритца, что позволяет найти компромисс между требуемой точностью и временем проектирования. Схема на основе компонентов может быть преобразована в физическую модель (фотошаблон или шестигранные сетки), используемую для 3D-анализа микросистемы.

Недостатки схмотехнического нисходящего проектирования:

1. Ограниченный набор допустимых для использования компонентов, которые хранятся в пользовательской библиотеке.
2. Невозможность получения топологии компонента или системы до тех пор, пока весь проект не будет закончен.
3. Параметры системы и компонентов не известны до последнего шага и, следовательно, очень трудно оптимизировать весь проект. Декомпозиция проекта или синтез повторяется для каждого уровня иерархии и разработчик не всегда может понять идет ли оптимизация в правильном направлении. Для того, чтобы уменьшить количество проектных итераций, проектировщик должен использовать концепцию «замыкания метрики», в которой различные значения параметров из нижних уровней абстракции используются для описания проекта на более высоких уровнях абстракции. В этом случае можно оценить оптимизированные значения параметров на более низком уровне абстракции в течение следующей итерации проектирования на более высоком уровне абстракции. К сожалению, концепцию «замыкания метрики» трудно реализовать на практике из-за высокой сложности реальных проектов.
4. Модели нестандартных компонентов произвольной геометрической формы имеют значительную сложность.
5. Физические эффекты на уровне упаковки кристалла в корпус, такие как влияние вязкоупругих характеристик полимерных материалов и эффекты формирования столбиковых выводов, трудно учесть при нисходящем схмотехническом проектировании на основе компонентов.

Восходящая методология проектирования на основе топологии (layout-based design) позволяет в первую очередь выполнить разработку компонентов и сохранить их в библиотеке для использования на следующем более высоком уровне абстракции. Двумерная модель топологии микросистемы формируется с помощью 2D моделей библиотечных компонентов и затем преобразуется в трехмерную модель путем моделирования технологических процессов изготовления МЭМС. Полученная 3D твердотельная модель микросистемы подвергается дискретизации и анализу с использованием методов 3D конечных / граничных элементов (FEA / BEA). Типовые 3D модели МЭМС могут иметь от 100 000 до 1 миллиона степеней свободы, в то

время как системные симуляторы не предназначены для обработки таких сложных проектов. Поэтому для представления микроструктуры в системе моделирования используются приближенные модели с сосредоточенными параметрами. Это позволяет уменьшить вычислительные и временные затраты моделирования, однако может привести к ошибкам при анализе физических процессов, происходящих в механических элементах микросистемы (например, при анализе возникающих напряжений в стержнях и эффекта левитации подвижных электродов из-за электростатического отталкивания). В процессе проектирования трехмерная твердотельная модель преобразуется в системную модель.

Достоинством восходящей методологии проектирования является связь проектной идеи с технологией изготовления микросистемы. Недостатком описанного подхода является то, что библиотеки должны включать в себя все возможные компоненты со всеми возможными параметрами, и они должны быть оптимизированы для любых настоящих и возможных будущих приложений. Это очень сложная и бесконечная задача, поскольку очень трудно предвидеть на нижнем уровне абстракции все потребности следующего более высокого уровня абстракции.

Методология деления пополам (*meet-in-the-middle*) дает возможность использовать преимущества обеих рассмотренных выше методологий нисходящего и восходящего проектирования, а также свести к минимуму их недостатки. Такой подход удобен тем, что стандарты проектирования и средства САПР на нижних уровнях абстракции достаточно хорошо развиты, в отличие от системного уровня. В *meet-in-the-middle* методологии применяется нисходящее проектирование на высших уровнях абстракции и восходящее – на нижних уровнях. Важной особенностью данного подхода является место перехода от одного стиля проектирования к другому. Так например, при проектировании цифровых систем на кристаллах *meet-in-the-middle* проектирование начинается с модели вычислений, на основе которой выполняется синтез системной платформы с виртуальными процессорными и коммуникационными компонентами, которые затем синтезируются с помощью RTL компонентов из RTL библиотеки. Процессорные и коммуникационные компоненты могут содержать IP-модули в виде списков соединений (*netlists*) RTL компонентов. Каждый RTL компонент имеет свою собственную структуру и топологию, полученные путем применения восходящей методологии. В процессе проектирования RTL компоненты объединяются путем их размещения и трассировки, формируя цифровую систему на кристалле.

На алгоритмическом уровне выполняется разработка алгоритмического описания процедур преобразования информации в МЭМС от момента ее получения до представления полученных результатов, генерируются стандартные и специализированные процессоры или специальные

аппаратные компоненты, такие как контроллеры памяти, арбитры, мосты, маршрутизаторы и компоненты интерфейса.

На системном уровне разрабатываются стандартные или встроенные системы, состоящие из процессоров, памяти, шин, компонентов МЭМС.

Новый класс численных алгоритмов на основе методов сокращения подпространств Крылова/Арнольди разработан для преобразования моделей анализа конечных элементов в модели с произвольным количеством степеней свободы. Модели FEA / BEA могут быть сведены к эффективным компактным моделям системы (моделям пониженного порядка – Reduced Order Models, ROMs), которые позволяют выполнять моделирование микроустройств с погрешностью 2-5% при сокращении времени расчета в 100-1000 раз по сравнению с полной численной моделью. Модели ROM точно описывают поведение устройства с точки зрения механической, тепловой, электрической энергий и могут быть преобразованы в модели описания аппаратуры на языках HDL (Hardware Description Languages).

Верификация MEMS отличается от верификации интегральных схем, в процессе которой выполняется проверка соответствия схемного описания топологии проекта с помощью инструмента проверки правил проектирования Design Rule Check (DRC). Инструмент DRC, дополненный средствами анализа кривых Безье и произвольной 3D геометрии компонентов, может быть использован для верификации МЭМС.

Верификация – процесс определения технического состояния изделия на различных стадиях его проектирования путем сравнения со спецификацией на основе моделирования тестовых или функциональных воздействий.

При проектировании МЭМС, верификация – это процесс определения технического состояния изделия путем сравнения результатов схемотехнического моделирования (методология сверху вниз) с результатами 3D моделирования (методология снизу вверх). Верификация МЭМС включает в себя сопоставительный анализ схемы, 3D конечных элементов и моделей ROM.

1.3. Виды неисправностей электронных компонентов МЭМС

Подходы к моделированию неисправностей, диагностированию, тестированию и обеспечению отказоустойчивости цифровых схем имеют важное значение для создания надежных изделий. С развитием VLSI технологии существенно увеличилось количество компонентов, размещаемых на одном кристалле и, как результат – возросла степень интеграции и количество неисправностей.

В процессе проектирования цифрового изделия необходимо периодически проверять (путем тестирования) корректность функционирования схемы и отсутствие неисправностей, а также обеспечить корректную работу схемы в случае появления неисправностей (отказоустойчивость).

Под неисправностью схемы понимают физический дефект одного или более компонентов схемы. Различают постоянные и нерегулярные неисправности. Постоянные (жесткие) неисправности могут быть следствием разрушения или износа компонента. Нерегулярные (мягкие) дефекты проявляются в определенные промежутки времени и могут быть кратковременными (transient) и перемежающимися (intermittent).

Неисправности могут быть логическими и параметрическими. Логическая неисправность изменяет булеву функцию, которая реализуется схемой. Параметрическая – изменяет значение параметра схемы (ток, напряжение). К параметрическим дефектам относят неисправности задержки, связанные с различным временем прохождения сигнала через логические вентили, что приводит к гонкам (состязаниям) сигналов.

Моделирование большого количества физических дефектов может быть основано на использовании одной модели логической неисправности, что позволяет существенно уменьшить сложность моделирования. Модель логической неисправности не зависит от технологии имплементации проекта, а тесты, разработанные для обнаружения логической неисправности, могут применяться также и для выявления физических дефектов.

Модель логической неисправности может быть явной или неявной. Явная модель неисправности определяет пространство неисправностей, в котором каждая неисправность может быть идентифицирована, а неисправности, подлежащие анализу, могут быть явно описаны. Явная модель неисправности практически применима, если ее размерность не является слишком большой. Неявная модель описывает пространство неисправностей путем совокупной идентификации неисправностей определенного типа их характеристическими признаками. Моделирование неисправностей тесно связано с моделированием системы. Неисправности, определяемые в сочетании со структурной моделью, относятся к структурным неисправностям, проявляющимся в изменении структуры межсоединений компонентов. Функциональные неисправности определяются в сочетании с функциональной моделью, так например, функциональная неисправность может проявляться в изменении таблицы истинности компонента или искажении RTL операции.

Классы неисправностей. Существует три класса логических неисправностей: константная неисправность (stuck-at-fault), мостиковая неисправность (bridging fault) и неисправность задержки (delay fault).

Наиболее распространенная модель константной неисправности – одиночная константная неисправность. Сущность модели заключается в том, что неисправность логического вентиля приводит к «залипанию» логического 0 (константа 0, s-a-0) или логической 1 (константа 1, s-a-1) на одном из его входов или выходов.

Модель константной неисправности также используется для представления кратных неисправностей в схеме. При этом предполагается, что более, чем на одной линии схемы имеется константная неисправность s-a-0 или s-a-1.

Другими словами, совокупность константных неисправностей существует в схеме в одно и то же время. Разновидностью кратной неисправности является однонаправленная неисправность – когда все конститuenty (составляющие части) неисправности представляют собой s-a-0 или s-a-1, но не обе одновременно.

Модель константной неисправности не является эффективной при моделировании СБИС (Very Large Scale Integrated, VLSI), построенных по CMOS технологии. Неисправности в CMOS схемах не обязательно представляют собой логические дефекты, которые могут быть описаны моделью константной неисправности. Дефекты CMOS-схем отображаются также моделями устойчивых обрывов транзисторов SOP (stuck-open) и устойчивых замыканий транзисторов SON (stuck-on) [9].

Мостиковые неисправности типа «короткое замыкание» представляют собой постоянные дефекты, которые не могут быть смоделированы константной неисправностью. Короткое замыкание возникает, когда две или более сигнальных линий схемы электрически связаны друг с другом. Мостиковые неисправности вентильного уровня классифицируются следующим образом: входные – вызваны коротким замыканием входов логического элемента, неисправности типа обратной связи – вызваны замыканием входной и выходной линий, а также неисправности без обратной связи, которые не относятся к первым двум типам. В теории моделирования мостиковых неисправностей делается предположение, что вероятность замыкания более двух линий является низкой и логика межсоединений реализуется в виде связей. Мостиковая неисправность в положительной логике возникает в том случае, когда ее поведение описывается проводным AND (0 является доминантным логическим значением), и в отрицательной логике – когда ее поведение описывается проводным OR (1 является доминантным логическим значением).

Неисправности задержки [10, 11]. Небольшое количество дефектов, которые могут вызвать разрывы и короткие замыкания в схеме, имеют достаточно высокую вероятность появления из-за наличия отклонений параметров производственного процесса. Дефекты могут также приводить к нарушениям временных параметров схемы без изменения логики ее работы: задержка переключения сигнала из 0 в 1 и наоборот. Существует два вида неисправностей задержки: неисправность задержки вентиля и неисправность задержки пути. Задержка вентиля используется для моделирования дефектов, при которых время прохождения сигнала через вентиль превышает предельно-допустимое. Данная модель может быть использована только для изолированных, не транспортируемых дефектов, например, несколько малых задержек. Модель задержки пути может быть использована как для изолированных, так и для транспортируемых дефектов. При этом предполагается, что неисправность проявляется в случае, если задержка распространения сигнала вдоль линии схемы превышает допустимое значение.

Кратковременные и перемежающиеся неисправности рассматриваются как временные дефекты. Основная часть неисправностей цифровых схем вызвана именно временными дефектами, которые характеризуются сложностью выявления и устранения. Кратковременные дефекты являются неповторяющимися и вызываются, как правило, флуктуациями напряжения питания или воздействием радиационного излучения. Они являются основной причиной отказа элементов памяти систем на кристаллах.

Перемежающиеся неисправности могут появляться в результате нарушения межсоединений, применения дефектных компонентов, воздействия внешних факторов (температура, влажность, вибрация) или быть следствием ошибок проектирования. Перемежающиеся неисправности возникают случайным образом и моделируются с помощью вероятностных методов (Марковские модели).

1.4. Модели функциональных неисправностей

Все более широкое использование программно-аппаратных систем в критических приложениях привело к повышению значимости верификации и тестирования программных и аппаратных модулей. В настоящее время существует ряд проблем верификации, связанных с высокой сложностью программно-аппаратных приложений и их гетерогенной структурой. Стоимость верификации системы увеличилась до такой степени, что иногда даже превышает стоимость проекта. Формальные методы верификации позволяют проверить функциональность с помощью формальных методов (проверка моделей, проверка эквивалентности, автоматическое доказательство теорем). Для управления сложностью задачи верификации, предложены методы, основанные на симуляции (эмуляции) описания системы заданной входной последовательностью.

Функциональные неисправности искажают пространство состояний цифрового изделия, представленное спецификацией. Дефект проектирования представляет собой неправильную деталь проекта, сформированную разработчиком. Дефекты проектирования являются следствием либо синтаксических (семантических) ошибок в описании устройства или фундаментального непонимания функциональности, описанной проектной спецификацией. Количество потенциальных дефектов проектирования может быть слишком большим, чтобы с ними можно было бороться автоматически или вручную, поэтому необходимо применять способы уменьшения сложности проекта без ущерба для точности результатов. Модель проектной неисправности описывает поведение некоторого множества дефектов проектирования. Модель функциональной неисправности описывает физические и проектные дефекты аппаратных и программных модулей. Модель функциональной неисправности можно оценить точностью моделирования проектных дефектов и эффективностью (количеством обнаруживаемых неисправностей схемы).

Большинство аппаратных систем разрабатываются на основе методологии проектирования сверху вниз, которая начинается с поведенческого описания

системы. Как результат, большинство моделей функциональных неисправностей являются моделями поведенческого или алгоритмического уровня. Существующие модели функциональных неисправностей могут быть классифицированы по стилю поведенческого описания, на котором они базируются. Системное поведение описывается на языках программирования (System C) или описания аппаратуры (VHDL, Verilog) и преобразуется во внутренний формат для использования в процессе симуляции.

1.4.1. Текстовые (семантические) модели неисправностей

Текстовая (семантическая) модель неисправностей используется для исходного текстового поведенческого описания проекта [12]. Простейшей текстовой моделью является метрика покрытия инструкций (statement coverage metric), используемая при тестировании программного обеспечения, которая связывает потенциальную ошибку с каждой строкой кода, и требует, чтобы каждый оператор поведенческого описания выполнялся во время тестирования. Эта модель не очень эффективна, поскольку количество возможных неисправностей равно числу строк кода. Ограничение точности покрытия инструкций позволяет в сочетании с другими моделями неисправностей повысить эффективность тестирования.

Ряд моделей функциональных неисправностей базируются на обходе путей графа потоков управления (Control-Data Flow Graph, CDFG), описывающего поведение системы [12]. Ранние модели неисправностей CDFG основывались на покрытии ветвей и путей графа. Покрытие ветвей предполагает, что множество всех проверяемых путей графа CDFG охватывает два направления реализации всех бинарных условий. Покрытие ветвей широко используется при тестировании программного и аппаратного обеспечения, однако использование только данной модели не позволяет получить полную гарантию корректности кода.

Метрика покрытия путей является более эффективной по сравнению с метрикой покрытия ветвей, поскольку она отражает количество путей графа потоков управления. Предполагается, что дефект связан с некоторым путем графа потоков управления и, следовательно, для гарантированного обнаружения всех неисправностей должны быть выполнены все пути потока управления. Количество путей управления может быть бесконечным, если граф CDFG содержит цикл. Поэтому, метрика обхода путей может быть ограничена длиной пути. Поскольку общее число путей потока управления растет экспоненциально с количеством условных операторов, можно выбрать подмножество всех путей потока управления, необходимое и достаточное для тестирования. Одним из критериев выбора пути может быть базисный набор путей или подмножество путей, которые линейно независимы и могут образовывать любой другой путь. При тестировании потоков данных появление каждой переменной рассматривается либо как описание переменной, либо как ее использование. При выборе пути рассматриваются такие, которые связывают определение переменной с ее использованием. Критерии тести-

рования потоков также применяются для проверки поведенческого описания аппаратных модулей.

Большинство моделей неисправностей графа потоков управления рассматривают пути без ограничения значений переменных и сигналов. В противоположность им, существуют модели неисправностей, ориентированные на переменные/сигналы, которые включают более жесткие ограничения на величину сигнала в целях обнаружения неисправностей. Методика анализа домена при тестировании программного обеспечения рассматривает не только путь потока управления, но и значения переменных и сигналов во время выполнения. Домен представляет собой подмножество пространства входных элементов программы, в котором каждый элемент активизирует выполнение программы по некоторому пути. Неисправность домена вызывает выполнение программы, следствием которого является переход в неправильную область. Данный метод может быть также применен для тестирования аппаратного обеспечения.

Многие модели неисправностей графа потоков управления содержат требования к активизации неисправностей не зависимо от значения наблюдаемости. Для устранения этого недостатка предложены поведенческие модели неисправностей, применимые для тестирования software и hardware модулей. Подход ОССОМ [12] базируется на добавлении неисправностей, называемых тегами, к каждому определению переменной, представляющих положительное или отрицательное смещение от правильного значения сигнала. Знак ошибки известен, но величина – нет. Анализ наблюдаемости вдоль пути потока управления делается вероятностно, с использованием алгебраических свойств операций и данных моделирования. Тег будет распространяться с помощью поведенческой операции, если будут выполнены два условия: 1) совпадет знак; 2) другие входы в процессе выполнения операции не контролируются. Разработан также точный метод определения наблюдаемости, в котором константная неисправность вводится на внутренних переменных, и ее распространение обеспечивается поведением объекта. Поскольку анализ наблюдаемости является точным, вычислительная сложность при этом возрастает.

1.4.2. Мутационные модели неисправностей

Мутационное тестирование основано на искусственном внесении неисправностей в код программы и применяется для тестирования программного и аппаратного обеспечений [12]. Основная идея заключается в имитации типичных ошибок программиста и создания специальных тестов для их выявления (тестов, которые бы обнаруживали неисправности, если бы они присутствовали). Неисправности вводятся в оригинальную программу и создается много неисправных версий программы. Каждая из них содержит одну ошибку. Ошибочные программы называются мутантами оригинальной программы. Целью генерации тестов является разграничение оригинальной программы и всех ее мутантов. Оригинальная программа и все ее мутации тес-

тируются на одном и том же наборе тестов. Если на этом наборе подтверждается правильность программы и выявляются все ошибки в программах-мутантах, то оригинальная программа объявляется правильной. Если в некоторых мутантах не были выявлены все ошибки, то тестовый набор считают не полным, и он подлежит доработке. Набор мутационных операторов языка VHDL включает изменения следующих объектов: арифметические операторы, определение и изменение абсолютного значения и константы, логические операторы, реляционные операторы, добавление унарного оператора. Каждый оператор представляет определенный класс неисправностей. Все возможные изменения в программе не могут рассматриваться из-за их непомерно большого количества. Изменения могут быть ограничены до приемлемого набора на основе двух гипотез: эффект сцепления и компетентный программист. Эффект сцепления гласит, что сложные неисправности могут сочетаться с простыми неисправностями, таким образом, тестовый набор, который обнаруживает все простые ошибки в программе, будет обнаружить также и сложные неисправности. Гипотеза «компетентный программист» утверждает, что компетентный программист стремится писать программы, которые практически являются правильными. Другими словами, программа, написанная компетентным программистом может быть неправильной, но она будет отличаться от правильной версии относительно простыми ошибками. Недостатком мутационных моделей является локальный характер мутаций, что ограничивает применение моделей для описания большого набора дефектов проектирования.

1.4.3. Модели неисправностей конечного автомата

Конечные автоматы (КА) являются классическим способом описания поведения последовательностных схем, и для них определены модели неисправностей [12]. Наиболее распространенной является модель покрытия состояний, основанная на требовании покрытия всех возможных состояний и выполнения всех возможных переходов в процессе тестирования. Дефекты КА не выводят его за пространство состояний, заданное спецификацией. Проблема использования моделей неисправностей конечного автомата заключается в высокой сложности решения задачи тестирования, которая обусловлена большой размерностью пространства состояний типовой вычислительной системы. Решением указанной проблемы может быть выявление подмножества состояний конечного автомата, имеющих решающее значение для его корректного функционирования. Модели расширенного конечного автомата (Extended Finite State Machine, EFSM) и машины контроля (Extracted Control Flow Machine, ECFM) позволяют уменьшить конечный автомат путем его разделения на пространство состояний и пространство данных. Уменьшенный конечный автомат генерируется путем проецирования оригинального конечного автомата на множество состояний, имеющих наибольшее значение для процесса валидации.

1.4.4. Модель исправного поведения объекта

Модель объекта диагностирования – это совокупность гетерогенных компонентов, взаимосвязанных во времени и пространстве, с заданной адекватностью описывающих некоторый процесс или явление. Модель может быть представлена в аналитической, табличной, векторной, графической или другой форме и задана в явном или неявном виде.

Явная модель объекта диагностирования состоит из описаний его исправной и всех неисправных модификаций. Неявная модель содержит описание исправного объекта, модели его физических неисправностей и правила получения по ним все неисправных модификаций объекта. Универсальной математической моделью объекта диагностирования является таблица функций неисправностей (ТФН). Каждое неисправное состояние объекта диагностирования соответствует одной неисправности (одиночной или кратной) из заданного класса неисправностей. Недостатком ТФН являются ее большие размеры. Модель дискретной системы может быть представлена в виде таблицы истинности, логической сети, альтернативного графа, эквивалентной нормальной формы представления булевых функций, таблицы переходовых выходов многотактной схемы. Выбор модели влияет на глубину и трудоемкость процесса диагностирования.

1.5. Основные понятия, термины и определения состояний МЭМС и свойств надежности

Термины и определения основных понятий, используемых в теории надежности, представлены в [13]. МЭМС характеризуются выходными параметрами, допустимые значения которых в процессе эксплуатации описаны в нормативно-технической (стандарты, технические условия) и/или конструкторской (проектной) документации.

ДСТУ 2860-94 визначає надійність як властивість об'єкта зберігати у часі в установлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування. Надійність є комплексною властивістю, що залежно від призначення об'єкта і умов його застосування, може містити в собі безвідмовність, довговічність, ремонтопридатність та збережуваність чи певні поєднання цих властивостей [13].

Безвідмовність – властивість об'єкта виконувати потрібні функції в певних умовах протягом заданого інтервалу часу чи наробітку [13].

Довговічність – властивість об'єкта виконувати потрібні функції до переходу у граничний стан при встановленій системі технічного обслуговування та ремонту [13].

Ремонтотпридатність – властивість об'єкта бути пристосованим до підтримання та відновлення стану, в якому він здатний виконувати потрібні функції за допомогою технічного обслуговування та ремонту [13].

Збережуваність – властивість об'єкта зберігати в заданих межах значення параметрів, що характеризують здатність об'єкта виконувати потрібні функції, під час і після зберігання та (чи) транспортування [13].

МЭМС может находиться в следующих технических состояниях: исправное, неисправное, работоспособное, неработоспособное и предельное.

МЭМС в исправном состоянии соответствует всем требованиям нормативно-технической и/или конструкторской (проектной) документации і здатна виконувати усі задані функції.

Если хотя бы по одному из требований система не соответствует нормативно-технической и/или конструкторской (проектной) документации, она находится в неисправном состоянии. Система у несправному стані нездатна виконувати хоч би одну із заданих функцій.

МЭМС может находиться в неисправном состоянии и при этом быть работоспособной. Нарушение исправного состояния системы при сохранении работоспособного состояния называется повреждением.

Если значения всех параметров, характеризующих способность системы выполнять заданные функции, соответствуют требованиям нормативно-технической и/или конструкторской (проектной) документации, то считается, что система находится в работоспособном состоянии. Працездатний стан характеризується здатністю системи виконувати усі потрібні функції [13].

Состояние МЭМС, при котором значение хотя бы одного параметра, характеризующего способность выполнять заданные функции, не соответствует требованиям нормативно-технической и/или конструкторской (проектной) документации, называется неработоспособным. Непрацездатний стан характеризується нездатністю системи виконувати хоч би одну з потрібних функцій [13].

Відмова – подія, яка полягає у втраті системою здатності виконувати потрібну функцію, тобто у порушенні працездатного стану МЭМС [13].

Критерий отказа – признак или совокупность признаков неработоспособного состояния системы, установленных в нормативно-технической и/или конструкторской (проектной) документации.

Критерии отказов:

- 1) прекращение выполнения системой заданных функций (отказ функционирования); снижение качества функционирования по одному или нескольким выходным параметрам (производительность, мощность, точность) за пределы допустимого уровня (параметрический отказ);
- 2) искажения информации на выходе системы;
- 3) внешние проявления, связанные с наступлением или предпосылками наступления неработоспособного состояния (шум, вибрации, перегрев).

Дефект – каждое отдельное несоответствие системы установленным требованиям.

Повреждение – событие, заключающееся в нарушении исправного состояния системы при сохранении работоспособного.

Наробіток – тривалість чи обсяг роботи системи [13].

Наробіток до відмови – наробіток системи від початку експлуатації до виникнення першої відмови.

Наробіток між відмовами – наробіток системи від завершення відновлення її працездатного стану після відмови до виникнення наступної відмови.

Граничний стан – стан об'єкта, за яким його подальша експлуатація неприпустима чи недоцільна, або відновлення його працездатного стану неможливе чи недоцільне [13].

Предельное состояние обусловлено физической невозможностью дальнейшей эксплуатации системы, недопустимым снижением ее эффективности, нарушением требований безопасности и определяется установленным критерием предельного состояния.

Критерій граничний стану – ознака чи сукупність ознак граничного стану системи, встановлених нормативною та/або конструкторською (проектною) документацією [13].

Критерии предельных состояний:

- 1) отказ одной или нескольких составных частей МЭМС, восстановление или замена которых на месте эксплуатации не предусмотрены эксплуатационной документацией;
- 2) механический износ ответственных деталей (узлов) или снижение физических (химических) свойств материалов до предельно допустимого уровня;
- 3) снижение наработки на отказ (повышение интенсивности отказов) ниже (выше) допустимого уровня;
- 4) повышение установленного уровня текущих (суммарных) затрат на техническое обслуживание и ремонт или другие признаки, определяющие экономическую нецелесообразность дальнейшей эксплуатации.

Ресурс – сумарний наробіток системи від початку її експлуатації чи поновлення після ремонту до переходу в граничний стан [13].

Термін служби – календарна тривалість експлуатації об'єкта від початку чи її поновлення після ремонту до переходу в граничний стан.

Ресурсна відмова – відмова, внаслідок якої система досягає граничного стану.

1.4. Срок службы МЭМС

Для прогнозирования срока службы компонентов МЭМС применяется ускоренное тестирование, в процессе которого микроизделие подвергается высокому уровню рабочих нагрузок и сильному воздействию окружающей среды (высокая температура и напряжение), что позволяет ускорить появление отказа. Алгоритм ускоренного тестирования МЭМС приведен на рис. 1.3 [4].

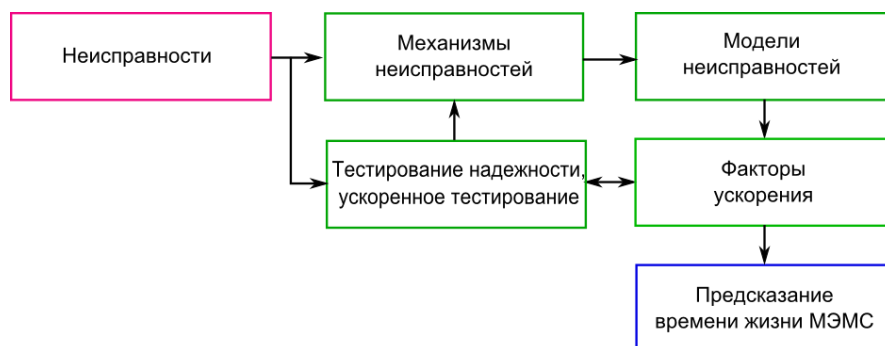


Рис. 1.3. Алгоритм предсказания срока службы МЭМС

2. МОДЕЛИ НАДЕЖНОСТИ МЭМС

2.1. Модели надежности МЭМС

Рассматриваются функция надежности, кумулятивная функция распределения отказов, функция распределения вероятности, функция интенсивности отказов и U-кривая надежности, используемые для прогнозирования срока службы компонентов МЭМС. Приводятся экспоненциальное и логарифмически нормальное распределение, распределение Вейбулла.

Под надежностью понимают вероятность того, что изделие выполняет определенные спецификацией функции при заданных режимах и условиях эксплуатации, хранения и транспортирования в течение нормативного срока службы.

Модели надежности МЭМС основаны на предположении о случайном характере отказов компонентов микросистем и предназначены для оценки показателей надежности МЭМС в процессе тестирования [4, 14]. Основными характеристиками надежности МЭМС являются [4, 14]:

1) Вероятность безотказной работы $R(t)$ – вероятность безотказного функционирования МЭМС на интервале времени $[0, t]$. Вероятность безотказной работы называют также функцией надежности [4].

2) Вероятность отказа $F(t)$ – вероятность возникновения отказа МЭМС на интервале времени $[0, t]$: $F(t) = 1 - R(t)$ (кумулятивная функция распределения отказов – cumulative failure distribution function, CDF).

Наиболее распространенным представлением закона распределения непрерывной случайной величины t является плотность распределения. Модель распределения наработки $f(t)$ – функция плотности вероятности (probability density function, PDF) в течение интервала времени от 0 до ∞ .

Функции CDF и PDF связаны отношением:

$$F(t) = \int_0^t f(t') dt'; \quad f(t) = \frac{d}{dt} F(t).$$

3) Интенсивность отказов $h(t)$ – условная плотность вероятности времени от возникновения отказа МЭМС при условии, что до момента t отказа не было:

$$h(t) = -\frac{dR(t)}{dt} / R(t) = \left| \frac{dF(t)}{dt} \right| / R(t).$$

Интенсивность отказов $h(t)$ определяется как отношение вероятности того, что отказ произойдет в следующем временном интервале, к надежности $R(t)$ (вероятности безотказной работы до этого временного интервала) [4]:

$$h(t) = \frac{f(t)}{1-F(t)} = \frac{f(t)}{R(t)}; \quad h(t) = -\frac{1}{R(t)} \frac{dR(t)}{dt} = -\frac{d}{dt} (\ln R(t)).$$

Интеграл от интенсивности отказов представляет собой кумулятивную интенсивность отказов:

$$H(t) = \int_0^t h(t') dt = -\ln R(t).$$

Интенсивность отказов $h(t)$ измеряется в единицах (время⁻¹). Поскольку в момент времени $t=0$ надежность изделия $R(0)=1$ (еще не было отказов), имеет место экспоненциальная зависимость надежности в интервале времени от 0 до t от интенсивности отказов в этом временном интервале:

$$R(t) = e^{-\int_0^t h(t') dt'}.$$

Важным количественным показателем надежности является наработка на отказ (mean time to failure, MTTF), определяющая время безотказной работы изделия (среднее время до первого отказа).

$$MTTF = \bar{t} \equiv \int_0^{\infty} t f(t) dt.$$

Модели надежности МЭМС могут основываться на предположении о дискретном изменении характеристик надежности и экспоненциальном характере изменения числа отказов в зависимости от времени тестирования и функционирования МЭМС.

2.2. Распределение надежности

Распределение отказов в течение срока службы продукта имеет важное значение для определения надежности МЭМС. Кривая зависимости интенсивность отказов от времени в течение срока службы имеет три характерных интервала, рис. 2.1: I — период приработки и отказов некачественных изделий; II — период нормальной эксплуатации; III — период старения (отказы вызваны износом деталей или старением материалов).

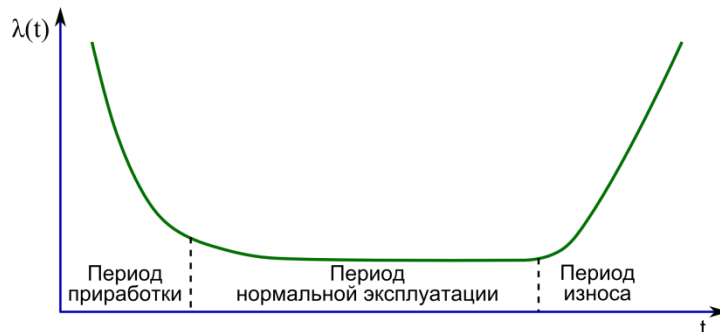


Рис. 2.1. Кривая зависимости интенсивности отказов от времени

В период приработки проявляются дефекты, обусловленные технологическими причинами, интенсивность отказов монотонно уменьшается.

Во время нормальной эксплуатации возникают, главным образом, внезапные отказы, интенсивность отказов остается приблизительно постоянной.

В период старения преобладают постепенные отказы, интенсивность отказов монотонно возрастает.

Законом распределения случайной величины называется соотношение, устанавливающее связь между возможными значениями случайной величины и соответствующими им вероятностями.

В теории и практике надежности чаще всего используются следующие законы распределения: нормальный (Гаусса), логарифмически нормальный, Вейбулла, экспоненциальный (показательный).

Для прогнозирования надежности в период нормальной эксплуатации изделий используется экспоненциальный закон распределения, описывающий распределение наработки на отказ сложных изделий, когда постепенные отказы еще не проявились и надежность характеризуется внезапными отказами. Эти отказы вызываются неблагоприятным стечением многих обстоятельств и поэтому имеют постоянную интенсивность.

Экспоненциальное распределение.

Экспоненциальное распределение вероятности безотказной работы является наименее сложным из всех моделей распределения случайной величины.

Интенсивность отказов $h(t)$ постоянна и равна λ . Функция надежности $R(t)$, кумулятивная функция распределения (CDF), функция распределения вероятностей (PDF) приведены ниже:

$$R(t) = e^{-\lambda t}; \quad (2.1)$$

$$F(t) = 1 - e^{-\lambda t}; \quad (2.2)$$

$$f(t) = \lambda e^{-\lambda t}. \quad (2.3)$$

Среднее время безотказной работы (наработка на отказ) определяется как величина, обратная интенсивности отказов λ $MTTF = \frac{1}{\lambda}$.

На рис. 2.2–2.4 представлены кумулятивная функция распределения отказов, функция распределения вероятности и функция интенсивности отказов.

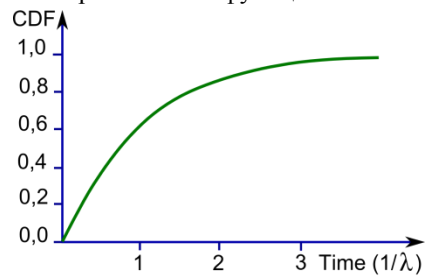


Рис. 2.2. Кумулятивная функция распределения отказов

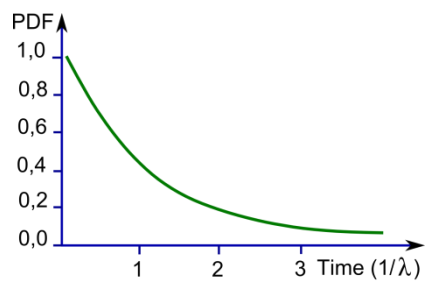


Рис. 2.3. Функция распределения вероятности

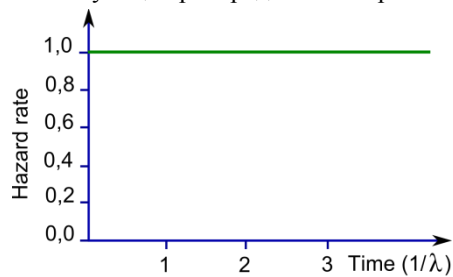


Рис. 2.4. Функция интенсивности отказов

Распределение Вейбулла.

Функция распределения Вейбулла используется для согласования кривых надежности различной формы. Функция Вейбулла может быть выражена несколькими способами [4]. Выражение распределения Вейбулла ниже описывает вероятность безотказной работы на интервале времени $[0, t]$. Выражение ниже вероятность выживания $R(T)$ между нулевой момент времени и времени

$$R(t) = e^{-\left(\frac{t-\gamma}{\alpha}\right)^\beta}.$$

Вид кривой надежности Вейбулла определяется тремя параметрами: β – форма, γ – положение, α – масштаб. Функция распределения Вейбулла может быть двухпараметрической или трехпараметрической в зависимости от того, начинаются ли отказы с нулевого момента времени. Если отказы начинаются с нулевого момента времени, то параметр положения, определяющий начальное время возникновения отказа, равен нулю, и экспоненциальное выражение распределения Вейбулла сводится к следующему:

$$R(t) = e^{-\left(\frac{t}{\alpha}\right)^\beta} = f(t)/h(t). \quad (2.4)$$

Если $\beta=1$, то выражение (2.4) принимает вид экспоненциальной зависимости (2.1) при $\alpha = 1/\lambda$.

Функция PDF двухпараметрического распределения Вейбулла представлена на рис. 2.5.

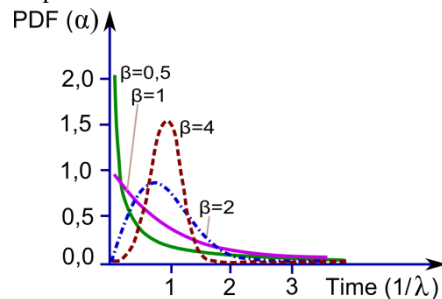


Рис. 2.5. Функция PDF двухпараметрического распределения Вейбулла в единицах α при различных значениях β

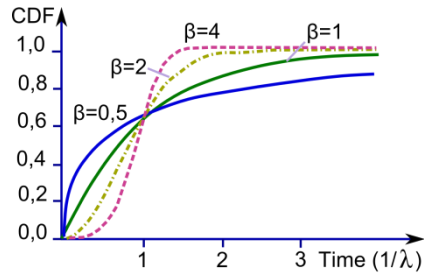


Рис. 2.6. Функция CDF распределения Вейбулла при различных значениях β

Распределение вероятности безотказной работы определяется выражением:

$$F(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^\beta}.$$

Согласно двухпараметрическому распределению Вейбулла интенсивность отказов $H(t) = \left(\frac{t}{\alpha}\right)^\beta$. В общем случае интенсивность отказов, которые носят

мгновенный характер определяется выражением: $h(t) = \left(\frac{\beta}{\alpha}\right)\left(\frac{t}{\alpha}\right)^{\beta-1}$.

Кривые интенсивности отказов для распределения Вейбулла приведены на рис. 2.7.

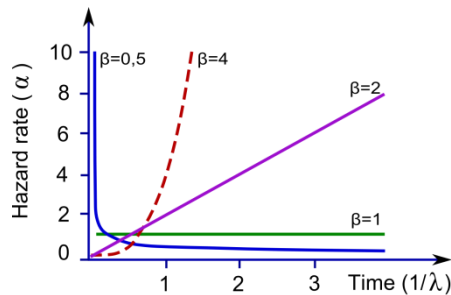


Рис. 2.7. Кривые интенсивности отказов для распределения Вейбулла

Логарифмически нормальное распределение.

Плотность вероятности нормально распределённой случайной величины t выражается формулой:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(t-m)^2}{2\sigma^2}\right)}, \text{ где } m \text{ и } \sigma - \text{параметры, определяющие положение}$$

и форму кривой: m – центр рассеивания случайной величины (математическое ожидание), характеризует положение кривой на оси абсцисс, σ – дисперсия, определяет форму кривой (при изменении дисперсии кривая будет сжиматься или растягиваться вдоль оси абсцисс).

При логарифмически нормальном распределении нормально распределенным является логарифм ($\ln t$) случайной величины t , а не сама эта величина. Логарифмически нормальное распределение во многом более точно, чем нормальное описывает наработку до отказа тех объектов, у которых отказ возникает вследствие усталости.

Функция плотности вероятности PDF при логарифмически нормальном (Гауссовом) распределении приведена на рис. 2.8 и определяется выражением:

$$f(t) = \frac{1}{\sigma t \sqrt{2\pi}} e^{\left(-\frac{(\ln(t) - \ln(T_{50}))^2}{2\sigma^2}\right)}, \text{ где } T_{50} - \text{средняя наработка до отказа; } \sigma$$

– среднее квадратичное (стандартное) отклонение времени безотказной работы, $\ln(T_{50})$ – математическое ожидание логарифма случайной величины.

Кумулятивная функция распределения CDF (рис. 2.9) $F(t)$:

$$F(t) = \int_0^t \frac{1}{\sigma t \sqrt{2\pi}} e^{\left(-\frac{(\ln(t) - \ln(T_{50}))^2}{2\sigma^2}\right)} dt;$$

$$F(t) = \Phi\left[\frac{\ln(t/\tau)}{\sigma}\right], \text{ где } \Phi(z) = \frac{1}{2} \left[1 + \operatorname{Erf}\left(z/\sqrt{2}\right)\right].$$

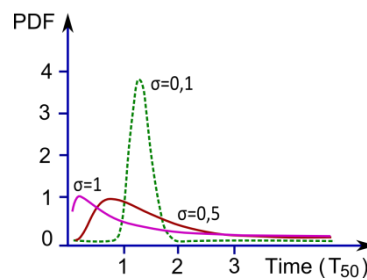


Рис. 2.8. Функция плотности вероятности при логарифмически нормальном распределении

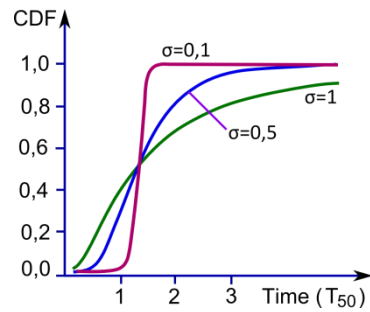


Рис. 2.9. Кумулятивная функция CDF при логарифмически нормальном распределении

Интенсивность отказов при логарифмически нормальном распределении возрастает с увеличением срока эксплуатации, см. рис. 2.10.

$$h(t) = \frac{f(t)}{1 - F(t)} = \frac{f(t)}{R(t)}$$

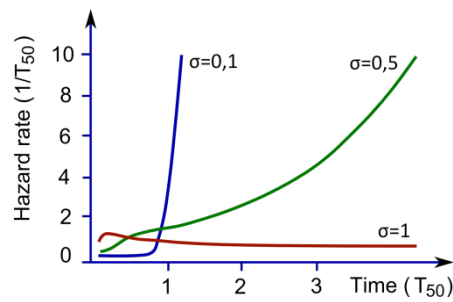


Fig. 2.10. Интенсивность отказов при логарифмически нормальном распределении

3. СТРУКТУРНЫЕ СХЕМЫ НАДЕЖНОСТИ МЭМС

3.1. Анализ модели надежности

Для получения достоверной информации о надежности устройства необходимо выполнять анализ модели надежности SRM (System Reliability Model), которая графически описывает систему в виде блок-схемы надежности (Reliability Block Diagram, RBD), дерева отказов (Fault Tree), дерева событий (Event Tree). Анализ позволяет определить узкие места проектируемой системы, которые могут привести к потере безотказности, долговечности, ремонтпригодности и сохраняемости. Модель SRM описывает структуру устройства как совокупности функциональных элементов и взаимосвязей между ними, включая аппаратные и программные компоненты устройства, а также приобретенные готовые стандартные элементы.

Блок-схема надежности RBD представляет собой графическое представление компонентов системы и их взаимосвязей, ориентированное на успех прохождения сигнала от входных портов к выходным, что свидетельствует о работоспособности системы. RBD позволяет определить следующие характеристики надежности: интенсивность отказов (мгновенную частоту отказов), среднее время наработки на отказ (MTBF), среднее время ремонта (MTTR).

При разработке модели системы необходимо определить работоспособное состояние и все виды отказов.

Построение RBD схем применяется для моделирования сложных систем с резервированием функций, а также систем, в состав которых входит встроенная инфраструктура сервисного обслуживания и восстановления работоспособности.

Исходными данными для построения RBD-схемы являются:

- результаты расчёта наработки на отказ (MTBF) для компонентов системы,
- времена циклов восстановления системы после отказа,
- вероятность успешного восстановления системы после отказа,
- интервалы между планово-предупредительными мероприятиями по техническому обслуживанию системы.

Различают последовательное, параллельное и смешанное соединение элементов в структурной схеме надежности.

3.1.1. Система параллельных элементов. При последовательном соединении отказ хотя бы одного элемента структурной схемы приводит к отказу всей системы в целом (рис. 3.1) [4, 15].



Рис. 3.1. Структурная схема надежности с последовательным соединением элементов

Если считать отказы элементов статистически независимыми, то вероятность безотказной работы системы выражается следующим образом [4]:

$$R_s(t) = r_1(t) \times r_2(t) \times \dots \times r_n(t) = \prod_{i=1}^n r_i(t), \text{ где } r_i(t) - \text{вероятность безотказной}$$

работы i -го элемента; $R_s(t)$ – вероятность безотказной работы системы.

Вероятность отказа системы на интервале $[0, t]$ равна

$$F_s(t) = 1 - R_s(t) = 1 - \prod_{i=1}^n [1 - f_i(t)].$$

Если надежность отдельного компонента системы не зависит от надежности остальных ее компонентов и соответствует экспоненциальному распределению, то надежность системы определяется выражением:

$$R_s(t) = \exp\{-h_1 t\} \times \exp\{-h_2 t\} \times \dots \times \exp\{-h_n t\}, \text{ где } h_1, h_2, \dots, h_n - \text{интенсивности}$$

отказов соответственно первого, второго и n -го элементов.

Для постоянного потока отказов с экспоненциальным распределением поток отказов системы описывается выражением: $h_s = h_1 + h_2 + \dots + h_n$. Т. е. Интенсивность отказов системы последовательно соединенных независимых элементов равна сумме интенсивностей отказов элементов.

Из приведенных формул следует, что при заданной вероятности безотказной работы элементов увеличение их числа резко снижает вероятность безотказной работы системы последовательно соединенных элементов.

3.1.2. Система параллельных элементов. Параллельным соединением элементов в структурной схеме надежности называется такое соединение, при котором система отказывает только при отказе всех n ее элементов [4, 15] (рис. 3.2).

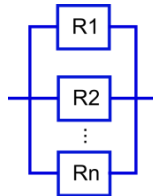


Рис. 3.2. Структурная схема надежности с параллельным соединением элементов

Вероятность отказа описывается выражением:

$$F_s(t) = f_1(t) \times f_2(t) \times \dots \times f_n(t) = \prod_{i=1}^n f_i(t) = \prod_{i=1}^n (1 - r_i(t)) .$$

Вероятность безотказной работы:

$$R_s(t) = 1 - F_s(t) = 1 - \prod_{i=1}^n (1 - r_i(t)) .$$

С учетом интенсивности отказов, при $h_i = \lambda$, выражение для вероятности безотказной работы примет вид:

$$P_s(t) = 1 - \prod_{i=1}^n (1 - \exp^{-\lambda t}) .$$

Из приведенных формул видно, что параллельное включение элементов является эффективным средством повышения надежности системы. Даже при весьма ненадежных элементах можно, наращивая число параллельных элементов, получить близкую к единице вероятность безотказной работы системы.

3.1.3. Система параллельно-последовательных элементов.

В сложных устройствах структурные схемы надежности могут содержать последовательные и параллельные структуры, как показано на рис. 3.3.

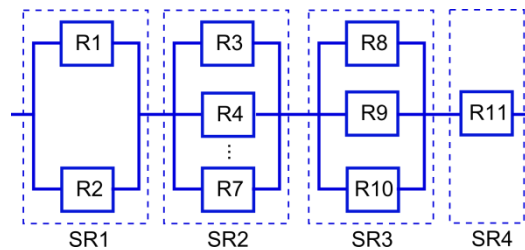


Рис. 3.3. Структурная схема надежности со смешанным соединением элементов

В этом случае для расчета надежности структурную схему разбивают на отдельные участки, на которых элементы системы соединены последовательно или параллельно. Структурная схема надежности преобразуется в последовательную структуру, показанную на рис. 3.4, где SR1, SR2, SR3, SR4 – вероятности безотказной работы соответственно первого, второго, третьего и четвертого последовательных участков.



Рис. 3.4. Преобразованная структура со смешанным соединением элементов

Тогда вероятность безотказной работы системы со смешанным соединением элементов будет равна:

$$R_s(t) = SR_1 \times SR_2 \times SR_3 \times SR_4.$$

В общем случае, для системы с k последовательными участками выражение для вероятности безотказной работы будет иметь вид:

$$R_s(t) = \prod_{j=1}^k SR_j(t),$$

где $SR_j(t)$ – вероятность безотказной работы j -го участка.

3.1.4. Система с мажоритарным резервированием

На рис. 3.5 представлена структурная схема надежности с мажоритарным резервированием. Избыточные компоненты запускаются с помощью переключателя в случае отказа основного элемента.

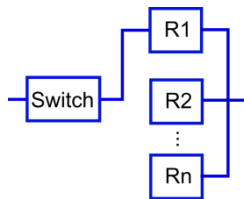


Рис. 3.5. Структурная схема надежности с мажоритарным резервированием

3.2. Анализ дерева неисправностей.

Анализ дерева неисправностей (отказов) FTA – это дедуктивный (причина-следствие), нисходящий метод, направленный на анализ последствий возникновения неисправностей и событий в сложной системе [16, 17]. Нежелательное последствие формирует корневую вершину (главное или конечное событие) дерева логики. Причинные факторы (ПФ) определяют дедуктивным образом, организуют логически и представляют наглядно с помощью древовидной схемы, которая отображает логическую связь ПФ с конечным событием.

Причинными факторами могут быть события, связанные с отказами оборудования или программного обеспечения, ошибками персонала, приводящими к возникновению нежелательного события.

Дерево неисправностей позволяет выявить элементы системы и события, которые приводят к данному конкретному отказу (рис. 3.6).

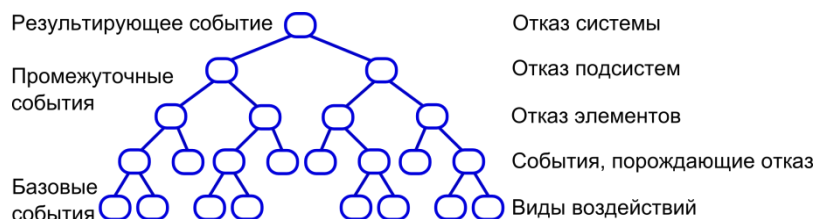


Рис. 3.6. Условная схема построения дерева отказов

Дерево неисправностей может применяться для качественного анализа системы – идентификации потенциальных причин и путей возникновения отказа (вершинного события), или для количественного – вычисления вероятности вершинного события, при наличии информации о вероятностях событий-факторов.

Недостатки дерева неисправностей [18]:

- реализация метода для сложных систем с большим количеством элементов требует значительных вычислительных затрат, поскольку увеличение глубины анализа приводит к геометрическому увеличению числа причинных факторов;
- дерево отказов использует только два состояния: рабочее и нерабочее;
- дерево отказов сложно построить для систем с резервированием и восстановлением, а также для анализа множественных отказов;
- дерево отказов описывает систему в определенный момент времени (в установившемся режиме), что не позволяет выполнить анализ последовательностей событий.

Дерево неисправностей позволяет на качественном уровне выявить возможные причины возникновения отказа и на количественном уровне определить вероятность конечного события на основании данных о вероятностях причинных событий.

Алгоритм построения дерева неисправностей [19]:

1. Определение конечного события.
2. Определение возможных непосредственных причин или типов отказа, приводящих к конечному событию.
3. Анализ каждой причины (или типа отказа) п.2 для определения причинных факторов, их вызывающих.
4. Поэтапное выявление нежелательного функционирования системы последовательно до более низких уровней до тех пор, пока дальнейший анализ не станет нецелесообразным (например, до уровня отказа отдельного компонента). Причинные факторы на самом низком уровне анализируемой системы называют базовыми событиями.
5. Расчет вероятности конечного события, при условии, что существует возможность определить вероятности базовых событий, и все входные данные

каждого логического элемента являются необходимыми и достаточными для наступления результирующего события.

6. Определение кратчайших путей возникновения конечного события и их влияния на конечное событие.

Дерево неисправностей представляет собой совокупность логических символов (вентилей) и символов событий, соединенных линиями. Логические символы связывают взаимодействующие события и условия в схеме дерева неисправностей. Диаграмма дерева неисправностей всегда изображается в направлении сверху вниз; объекты на нижнем уровне являются базовыми блоками событий. Классические вентили дерева неисправностей не имеют свойств (они не могут быть неисправными).

Обозначения логических символов приведены ниже [16]. Логический символ может иметь один или несколько входов и только один выход, или выходное событие.

Логический символ «И» (AND, схема совпадения) определяет наступление выходного события в том случае, если все входные события появляются одновременно (рис. 3.7). С точки зрения надежности системы, это означает: для того, чтобы произошел отказ системы (выход) все компоненты должны отказать (вход). Эквивалентной RBD схемой является простая параллельная конфигурация.



Рис. 3.7. Логический символ «И»

Пример использования логического вентиля «И» в дереве неисправностей системы при наличии двух компонентов А, В и эквивалентная RBD схема приведены на рис. 3.8. Для отказа системы необходимо, чтобы отказали оба компонента.

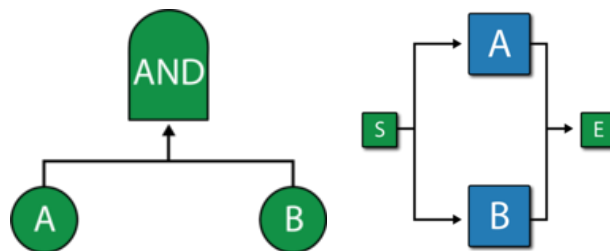


Рис. 3.8. Логический символ «И» и эквивалентная RBD схема

Уравнение надежности для любой конфигурации системы (рис. 3.8) имеет вид:

$$R_{\text{system}} = R_A + R_B - R_A \cdot R_B.$$

Логический символ "ИЛИ" (OR, схема объединения) определяет наступление выходного события в том случае, если имеет место любое из входных

событий (рис. 3.9). На рис. 3.10 показан фрагмент дерева неисправностей, в котором выходное событие происходит в том случае, если имеет место событие А или В. Два указанных события связаны логическим элементом ИЛИ. Если выходное событие представляет собой сбой (отказ) системы, а два входных события – отказы ее компонентов, то дерево неисправностей говорит о том, что отказ элементов А или В приводит к отказу системы.



Рис. 3.9. Логический символ «ИЛИ»

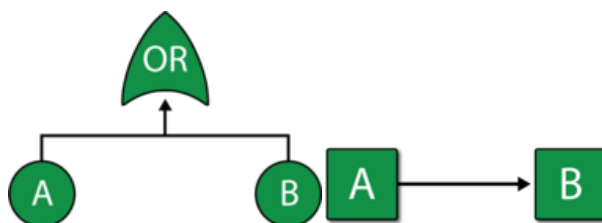


Рис. 3.10. Логический символ «ИЛИ» и эквивалентная RBD-схема

Эквивалентная RBD-схема для приведенного фрагмента имеет вид последовательной структуры (рис. 3.10).

Пример использования логического вентиля «ИЛИ» в дереве неисправностей системы при наличии трех компонентов А, В, С и эквивалентная RBD схема приведены на рис. 3.11.

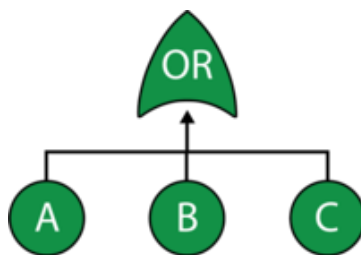


Рис. 3.11. Логический символ «ИЛИ» и эквивалентная RBD-схема для системы с тремя компонентами

Уравнение надежности для любой конфигурации системы (рис. 3.11) имеет вид:

$$R_{\text{system}} = R_A \cdot R_B \cdot R_C$$

Причинные связи, выраженные логическими символами И, ИЛИ, являются детерминированными, так как появление выходного события полностью определяется входными событиями.

Мажоритарный элемент «ИЛИ» (Voting OR Gate) определяет наступление выходного события, если происходят k или более выходных событий (рис. 3.12). С точки зрения надежности системы, это означает, что если какие-либо k из n компонентов выйдут из строя (вход), то произойдет отказ системы (выход).

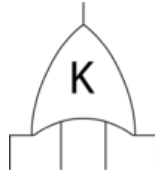


Рис. 3.12. Мажоритарный элемент «ИЛИ»

Пример использования логического вентиля «мажоритарный элемент ИЛИ» в дереве неисправностей системы при наличии четырех компонентов A, B, C, D и эквивалентная RBD схема приведены на рис. 3.13. Эквивалентная RBD схема представляет собой узловую вершину и подобна параллельной структуре k -из- n , но имеет одно отличие: в системе 2-из-4 произойдет отказ, если выйдут из строя любые 2 компонента; согласно RBD схеме система 3-из-4 будет находиться в исправном состоянии, если исправны любые 3-из-4 компонентов, и выйдет из строя, если откажут более одного компонента.

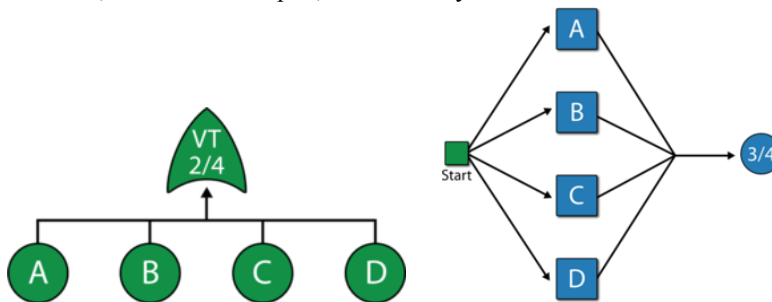


Рис. 3.13. Логический символ «мажоритарный элемент ИЛИ» и эквивалентная RBD-схема

Другими словами, дерево неисправностей определяет k -из- n отказов, необходимых для отказа системы, в то время как RBD-схема определяет k -из- n успехов для успеха системы.

Пример дерева неисправностей и RBD-схемы для системы с тремя компонентами A, B и C приведен на рис. 3.14. Отказ системы произойдет, если любые два компонента выйдут из строя.

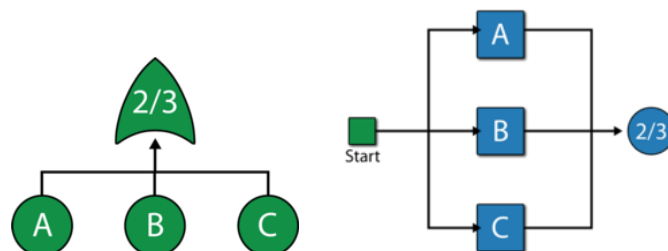


Рис. 3.14. Логический символ и RBD-схема для системы 2-из-3

Уравнение надежности для любой конфигурации системы (рис. 3.14) имеет вид (при допущении, что мажоритарный элемент «ИЛИ» не может выйти из строя):

$$R_{\text{system}} = -2 \cdot R_A \cdot R_B \cdot R_C + R_A \cdot R_B + R_A \cdot R_C + R_B \cdot R_C.$$

В противном случае (если возможен отказ мажоритарного элемента «ИЛИ»): уравнение изменяется следующим образом:

$$R_{\text{system}} = R_{\text{VOR}}(-2 \cdot R_A \cdot R_B \cdot R_C + R_A \cdot R_B + R_A \cdot R_C + R_B \cdot R_C),$$

где R_{VOR} – вероятность безотказной работы мажоритарного элемента «ИЛИ».

В дереве неисправностей мажоритарный элемент «ИЛИ» указывает на то, что если любые два компонента выйдут из строя, то произойдет отказ системы; в то время как узел в RBD-схеме указывает на то, что, если по крайней мере, два компонента исправны, то система будет функционировать исправно.

При построении дерева неисправностей и RBD-схем сложных объектов можно использовать различные комбинации логических вентилях (рис. 3.15).

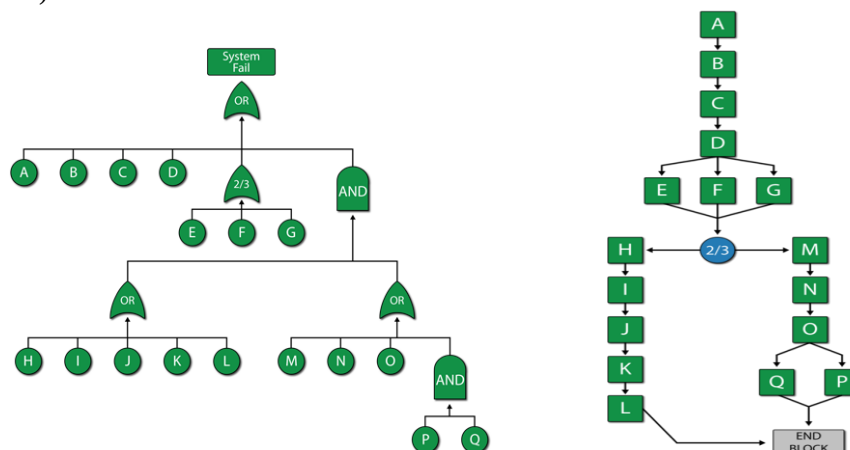


Рис. 3.14. Фрагмент дерева неисправностей и RBD-схемы с комбинацией логических вентилях

Кроме основных логических вентилях, рассмотренных выше, при построении деревьев неисправностей более сложных конфигураций с избыточностью используются также следующие элементы: принудительная последовательность (Sequence Enforcing), «И» с приоритетом (Priority AND).

Элемент «принудительная последовательность» изображается различными символами, например, символом «И», который определяет появление каждого события в определенной последовательности. Другими словами, на события накладываются ограничения в целях обеспечить их появление в определенной последовательности; выходное событие происходит, если все входные события выполнены в заданной последовательности. Сущность рассматриваемого логического вентиля идентична конфигурации с холодным резервированием, когда k резервных модулей находятся в режиме ожидания и до своего включения в работу не могут отказывать.

Запрещающий логический вентиль (Inhibit Gate) определяет ситуацию, когда происходит выходное событие в том случае, если происходят все входные события и дополнительное условное событие. Inhibit Gate – это логический элемент с дополнительным событием. В действительности запрещающий логический вентиль не предоставляет никаких дополнительных возможностей в области моделирования, но используется для описания ситуации, когда дополнительное событие также должно произойти.

В качестве примера рассмотрим ситуацию, когда при возникновении событий А и В, а также третьего (условного) события С происходит сбой системы. Можно построить дерево неисправностей с использованием логического элемента «И» (рис. 3.15, а) или запрещающего логического вентиля Inh (рис. 3.15, б), изображаемого в виде шестиугольника. При этом условное событие изображается справа от вентиля Inh и соединяется с ним направленной дугой.

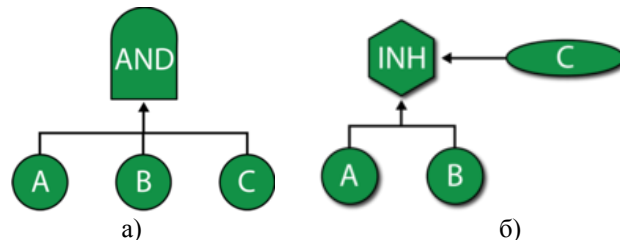


Рис. 3.15. Дерево неисправностей с использованием: а) элемента «И»; б) запрещающего элемента

Логический вентиль «И» с приоритетом определяет наступление выходного события в том случае, если все входные события происходят в определенной последовательности (рис. 3.16). Элемент «И» с приоритетом отличается от Sequence Enforcing тем, что для событий, которые должны произойти, задается приоритет (например, событие А должно произойти раньше, чем событие В).

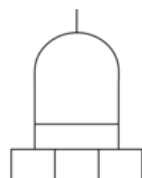


Рис. 3.16. Логический вентиль «И» с приоритетом

Переходной вентиль вход/выход используется для обозначения перехода от одного дерева неисправностей к другому или продолжения одного дерева на другом листе (рис. 3.17).



Рис. 3.17. Переходной вентиль

Логический вентиль XOR определяет возникновение выходного события в том случае, если происходит только одно входное событие (рис. 3.18). При наступлении более одного входного события выходное событие не наступает. В случае, когда ни одно входное событие не происходит, выходное событие также не наступает. С точки зрения надежности системы это означает, что двухкомпонентная система будет исправно функционировать, даже если произойдет отказ обоих компонентов. Если одновременный отказ обоих компонентов является недостижимым состоянием, в дереве неисправностей можно использовать логический вентиль ИЛИ.

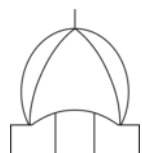


Рис. 3.18. Логический вентиль XOR

Традиционные деревья отказов используют различные формы представления событий (рис. 3.19): базовое (basic event), неразвитое (undeveloped event), внешнее (house event) и условное (conditional event). Базовое событие (отказ) идентично RBD блоку и изображается окружностью (рис. 3.19, а). Неразвитое событие идентично базовому, но может быть развито в отдельное дерево неисправностей (рис. 3.19, б). Внешнее событие может произойти или не произойти и имеет фиксированную вероятность возникновения от 0 до 1 (рис. 3.19, в). Используется для определения путей включения/выключения или функциональных/нефункциональных. Внешние события failed house (отказ) и working house (функционирование) могут иметь вероятность 0 или 1. Условное событие изображается эллипсом и задает условие, которое может быть применено к любым логическим вентилям (рис. 3.19, г).

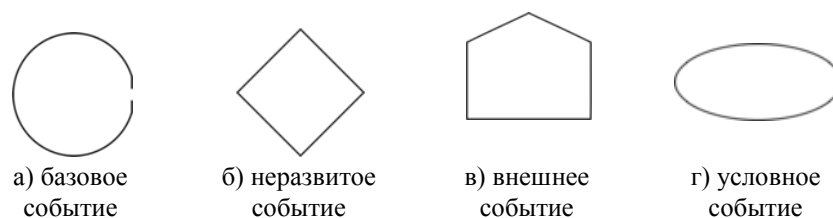


Рис. 3.19. Виды событий

3.3. Анализ дерева событий.

Анализ дерева событий (Event Tree Analysis, ETA) заключается в построении взаимоисключающих последовательностей событий, исходящих из основного события (аварийной ситуации), в целях определения вариантов (сценариев) развития его последствий [20]. Частота каждого сценария рассчитывается путем умножения частоты основного события на условную вероятность конечного события.

Метод построения дерева событий – это графический способ слежения за набором обстоятельств (отказ системы и внешние воздействия на нее), ведущих к неблагоприятному исходу (рис. 3.20). В отличие от метода анализа дерева неисправностей, целью моделирования методом построения дерева событий является установление последствий наступления нежелательного события.

В каждой технической системе есть способы, устройства, приспособления, предохраняющие систему от аварий и катастроф, которые называются элементами защиты. Элементы защиты системы могут сработать или не сработать. Все события, которые могут произойти после произошедшего основного события, соединяются причинно-следственными связями в зависимости от срабатывания или отказа элементов защиты системы. Как правило, ствол дерева располагается в левой части схемы. Стволом дерева является неблагоприятное событие (инцидент), из ствола расходятся ветви – возможные пути развития последствий инцидента. Верхние ветви дерева отражают развитие событий при срабатывании элемента защиты и называются ветвями срабатывания. Нижние ветви называются ветвями отказов. В общем случае элементы защиты могут функционировать независимо друг от друга. Вне зависимости от того, сработал ли какой-либо элемент защиты системы, другой может также сработать либо отказаться.

Метод построения дерева событий позволяет выполнить качественный и количественный анализ сценариев развития событий. В соответствии с теорией вероятностей для независимых событий вероятность реализации каждой последовательной цепи определяется произведением вероятностей каждого из событий цепи. Необходимым условием построения дерева событий является информация об отказах оборудования. Точность этих данных существенно влияет на точность расчетов финальных вероятностей.

дение отдельных условных вероятностей и частоты исходного события, при условии что различные события являются независимыми.

Результатом анализа может быть качественное описание потенциальных проблем в зависимости от комбинаций отдельных событий или количественные оценки частоты событий и их вероятностей.

Преимущества дерева событий заключаются в следующих возможностях:

- 1) отображение возможных сценариев развития ситуации в результате появления исходного события и определение влияния инцидента на функционирование системы;
- 2) учет фактора времени, взаимосвязи событий и «эффекта домино», моделирование которых в рамках дерева неисправностей затруднительно;
- 3) представление в графической форме последовательности событий.

Недостатки метода:

- 1) применение дерева событий для всестороннего анализа функционирования системы требует выявления всех возможных исходных событий, что увеличивает вычислительную сложность, и всегда существует вероятность невыявления отдельных значимых исходных событий;
- 2) метод ориентирован на использование только исправных и неисправных состояний системы; отложенные исправные состояния или события восстановления работоспособности не рассматриваются;
- 3) каждый путь в дереве обусловлен сочетанием событий, произошедших на предыдущих точках ветвлений, взаимосвязанные события не рассматриваются, что может приводить к недооценке риска.

3.4. Прогнозування надійності.

Державний стандарт ДСТУ 2861-94 рекомендує використання методів прогнозування надійності, викладених нижче [17].

Прогнозування надійності за кількістю елементів (лямбда-метод) – це переважно індуктивний (знизу нагору) метод, що використовується на стадіях технічної пропозиції та початкового проектування для оцінки приблизної інтенсивності відмов системи. Реалізація методу передбачає формування послідовної логічної структури елементів системи та визначення інтенсивності відмов відповідно до їх рівнів навантаження. Це дозволяє оцінити найбільш несприятливий варіант (найгірший випадок) поведінки системи. Якщо відомі стани резерву на вищих рівнях системи, то їх наявність може бути взята до уваги. Прогнозування надійності з використанням лямбда-методу дає істотне зниження показників надійності (середнього наробітку до відмови) при великій кількості елементів.

Марковський аналіз – це переважно індуктивний (знизу нагору) метод аналізу, який використовується для оцінки функціонально складних систем і стратегій технічного обслуговування та ремонту. Метод ґрунтується на теорії марковських процесів і передбачає визначення всіх станів та імовірностей переходу системи з одного стану в інший (частоти відмов або ремонту, час-

тоти події). Припущенням є незалежність частоти переходів від часу чи передісторії. Імовірності для елементів системи (компонентів, підсистем), що перебувають в конкретному функціональному стані у заданих точках або інтервалах часу, оцінюються за допомогою існуючих математичних моделей. Для проведення якісного аналізу та розрахунку надійності системи потрібно визначити всі можливі стани та побудувати діаграму переходів з одного стану в інший.

Імовірнісно-фізичний метод аналізу надійності передбачає використання моделей надійності, які мають конкретну фізичну інтерпретацію та пов'язують значення імовірності відмови і фізичного (визначального) параметра, що призводить до відмови. Кількісні оцінки показників надійності систем (елементів) визначають як на основі статистичної інформації про відмови, так і статистичної інформації про фізичні процеси, що призводять до відмов.

4. РЕЗЕРВИРОВАНИЕ НЕВОССТАНАВЛИВАЕМЫХ СИСТЕМ

4.1. Раздельное резервирование системы

Резервування – спосіб забезпечення надійності об'єкта за рахунок використання додаткових компонентів, надлишкових відносно основних елементів, мінімально необхідних для виконання потрібних функцій [13].

Дополнительная избыточность позволяет восстановить работоспособность системы при отказе одного или нескольких основных элементов.

Различают резервирование общее и раздельное. При общем резервировании дублируется объект (система) в целом, при раздельном – отдельные элементы системы.

Кратность резервирования k определяется как отношение количества резервных компонентов к числу резервируемых (основных) элементов системы. Кратность резервирования может быть целой и дробной. При резервировании с целой кратностью k представляет собой целое число, с дробной – дробное несокращаемое число. Например, $k = 4/2$ означает, что имеется четыре резервных компонента и два основных; $k = 2$ означает, что имеется два резервных элемента и общее число элементов равно трем.

По способу включения выделяют постоянное резервирование и резервирование замещением. При постоянном резервировании резервные компоненты подключены к основным в течение всего времени работы системы и функционируют в одинаковом с ними режиме. При резервировании замещением резервные элементы замещают основные после их отказа.

Если система состоит из не однотипных элементов, то применяется общий случай раздельного резервирования – резервирование по группам [21]. Обозначим через S_k число компонентов в k -й группе элементов системы. Если общее число групп равно g , а основная система содержит n последовательных элементов, то

$$\sum_{k=1}^g S_k = n, \quad 1 \leq g \leq n. \quad (4.1)$$

Допустим, общее число экземпляров k -ой группы элементов равно m_k . Отказ этой группы возникает в том случае, если возникнут отказы во всех m_k экземплярах. Тогда вероятность отказа $F_k(t)$ указанной группы равна

$$F_k(t) = \prod_{i=1}^{m_k} \left[1 - \prod_{j=N_{k-1}+1}^{N_k} r_{ij}(t) \right], \quad (4.2)$$

$$\text{где } N_l = \sum_{n=1}^1 s_n; l=1, \dots, g \quad (4.3)$$

$$N_0 = 0; N_g = n; N_k - N_{k-1} = s_k;$$

$r_{ij}(t)$ – вероятность безотказной работы элемента с номером j в последовательном соединении и с номером i в параллельном.

Выражение вероятности $R(t)$ безотказной работы системы имеет вид произведения вероятностей безотказной работы $1 - F_k(t)$ отдельных групп:

$$F(t) = \prod_{k=1}^g \left\{ 1 - \prod_{i=1}^{m_k} \left[1 - \prod_{j=N_{k-1}+1}^{N_k} r_{ij}(t) \right] \right\} \quad (4.4)$$

При $g=1$ имеет место общее резервирование системы, при $g=n$ – поэлементное.

4.2. Резервирование системы однотипных элементов

Если все элементы системы однотипны, вероятность безотказной работы любого элемента системы равна $r(t)$ и число последовательных элементов в каждой группе постоянно и равно $s = n/g$ (предполагается, что n кратно g). Тогда вероятность безотказной работы одной цепочки в каждой группе, состоящей из $s = n/g$ последовательных элементов, будет равна $r^s(t)$. Если общее число экземпляров каждой группы постоянно и равно m , то для каждой из групп вероятность безотказной работы равна $1 - [1 - r^s(t)]^m$. Вероятность $r_{mm}^{(g)}(t)$ безотказной работы системы на интервале $[0, t]$ равна вероятности безотказной работы g таких последовательных групп:

$$R_{mm}^{(g)}(t) = \{1 - [1 - r^s(t)]^m\}^g, \quad s = n/g. \quad (4.5)$$

Выражение (4.5) является частным случаем (4.4) при $s_k = s$, $m_k = m$, $r_{ij}(t) = r(t)$ для всех k, i, j . Кратность резервирования в этом случае является целой величиной, равной $m-1$.

Частные случаи:

$$1) \ g=1 \text{ (общее резервирование)} \quad (4.6)$$

$$2) \ g=n \text{ (резервирование по элементам)} \\ R_{mm}^{(n)}(t) = \{1 - [1 - r(t)]^m\}^n. \quad (4.7)$$

Если известна требуемая вероятность безотказной работы $R(t)$ резервируемой системы на интервале $[0, t]$, используя выражение (4.5 2.15) можно оп-

ределить необходимую для выполнения этого требования вероятность безотказной работы каждого элемента на указанном интервале времени:

$$r(t) = \{1 - [1 - R^{1/g}(t)]^{1/m}\}^{g/n}. \quad (4.8)$$

При общем резервировании ($g=1$)

$$r(t) = \{1 - [1 - R(t)]^{1/m}\}^{1/n}. \quad (4.9)$$

При резервировании по элементам

$$r(t) = 1 - [1 - R^{1/n}(t)]^{1/m}. \quad (4.10)$$

4.3. Дублирование

Вероятность безотказной работы системы при однократном резервировании (дублировании) при $m=2$ имеет вид:

$$R_{2n}^{(g)}(t) = \{1 - [1 - r^s(t)]^2\}^g. \quad (4.11)$$

При общем дублировании ($g=1$)

$$R_{2n}^{(1)}(t) = 1 - [1 - r^n(t)]^2, \quad (4.12)$$

при дублировании по элементам ($g=n$)

$$R_{2n}^{(n)}(t) = \{1 - [1 - r(t)]^2\}^n. \quad (4.13)$$

Вероятность безотказной работы каждого элемента, необходимая для обеспечения заданной вероятности безотказной работы системы $R(t)$, может быть определена из (4.8 2.18):

$$r(t) = \{1 - [1 - R^{1/g}(t)]^{1/2}\}^{g/n}. \quad (4.14)$$

При общем дублировании

$$r(t) = \{1 - [1 - R(t)]^{1/2}\}^{1/n}; \quad (4.15)$$

при дублировании по элементам

$$r(t) = 1 - [1 - R^{1/n}(t)]^{1/2}. \quad (4.16)$$

4.4. Резервирование замещением

Нагруженное резервирование однотипных элементов. При резервировании замещением резервные элементы могут находиться в одном из двух предельных режимов: нагруженный и ненагруженный. Резервные элементы замещают отказавшие при помощи переключающих устройств. Обозначим $r_{sw}(t)$ вероятность безотказной работы переключающего устройства, причем отказ переключателя обнаруживается только в момент отказа основного элемента. Переключающее устройство выполняет переключение по g груп-

пам системы n однотипных элементов при нагруженном режиме резервных элементов.

Вероятность безотказной работы основного комплекта одной группы элементов равна $[r(t)]^{n/g}$, а вероятность безотказной работы резервного комплекта этой группы с учетом переключающегося устройства равна $r_{sw}(t)[r(t)]^{n/g}$. Вероятность безотказной работы резервированной системы равна

$$R_{nn}^{(g)}(t) = \{1 - [1 - r^{n/g}(t)][1 - r_{sw}(t)r^{n/g}(t)]^{m-1}\}^g. \quad (4.17)$$

При общем резервировании

$$R_{nn}^{(1)}(t) = 1 - [1 - r^n(t)][1 - r_{sw}(t)r^n(t)]^{m-1}; \quad (4.18)$$

при резервировании по элементам

$$R_{nn}^{(n)}(t) = \{1 - [1 - r(t)][1 - r_{sw}(t)r(t)]^{m-1}\}^n \quad (4.19)$$

4.5. Нагруженное резервирование неоднотипных элементов

Формулу (4.17) можно обобщить на систему, состоящую из неоднотипных элементов. Эта формула, как нетрудно доказать, имеет вид

$$R(t) = \prod_{k=1}^g \{1 - [1 - \prod_{j=N_{k-1}+1}^{N_k} r_{ij}(t)] \prod_{i=2}^{m_k} [1 - r_{swk}(t) \prod_{j=N_{k-1}+1}^{N_k} r_{ij}(t)]\} \quad (4.20)$$

где $r_{swk}(t)$ – вероятность безотказной работы переключающего устройства, осуществляющего переключение на резерв k -й группы элементов системы.

4.6. Скользящее резервирование

Для резервирования нескольких одинаковых (или взаимозаменяемых) элементов системы одним или несколькими резервными применяется метод скользящего резервирования (рис. 4.1), позволяющий уменьшить количество резервных компонентов. Отказ системы произойдет, если число отказавших основных элементов превысит число резервных. Скользящее резервирование может быть как нагруженным, так и ненагруженным.

Пример. Система состоит из n однотипных последовательно соединенных элементов и m резервных, каждый из которых может при помощи переключающего устройства замещать любой отказавший основной элемент системы (рис. 4.1).

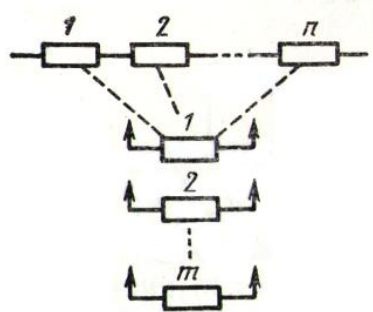


Рис. 4.1. Система последовательно соединенных элементов со скользящим резервированием

Основные и резервные элементы находятся в нагруженном режиме. Вероятность безотказной работы переключающего устройства $r_{sw}(t)$, вероятность безотказной работы любого элемента равна $r(t)$.

Система будет работоспособной до тех пор, пока из $n+m$ основных и резервных элементов работают безотказно не менее, чем n элементов. Использование резервного элемента связано с безотказной работой переключающего устройства, следовательно, вероятность безотказной работы резервного элемента с учетом переключения равна $r_{sw}(t)r(t)$. Вероятность безотказной работы системы со скользящим резервированием определяется по формуле:

$$R_{nm}^{(rov)}(t) = \sum_{i=n}^{n+m} \sum_{k=i-n}^m C_{i-k}^n C_k^m r^{i-k}(t) [1-r(t)]^{n-i+k} [r_{sw}(t)r(t)]^k [1-r_{sw}(t)r(t)]^{m-k} \quad (4.21)$$

Если ненадежность переключающего устройства можно не учитывать ($r_{sw}(t) \equiv 1$), то из (4.21) находим

$$R_{nm}^{(rov)}(t) = \sum_{i=n}^{n+m} \sum_{k=i-n}^m C_{i-k}^n C_k^m r^i(t) [1-r(t)]^{m+n-i} = \sum_{i=n}^{n+m} C_i^{m+n} r^i(t) [1-r(t)]^{m+n-i}; \quad (4.22)$$

Поскольку $\sum_{k=i-n}^m C_{i-k}^n C_k^m = C_i^{m+n}$, то формула (4.22) преобразуется к виду:

$$R_{nm}^{(rov)}(t) = 1 - \sum_{i=0}^{n-1} C_i^{m+n} r^i(t) [1-r(t)]^{m+n-i}. \quad (4.23)$$

4.7. Частные случаи

Если система состоит только из одного элемента ($n=1$), то $(m-1)$ -кратное скользящее резервирование при безотказном переключающем устройстве совпадает со случаем включения m параллельных элементов. Из (4.23) при $n=1$ получаем

$$R_{m1}^{(rov)}(t) = 1 - [1 - r(t)]^m.$$

Если m резервных элементов используются по методу скользящего резервирования в системе, состоящей из двух последовательно соединенных элементов, то используя выражение (4.22) при $n=2$ имеем:

$$R_{m2}^{(rov)}(t) = 1 - C_1^{m+2} r(t) [1 - r(t)]^{m+1} - [1 - r(t)]^{m+2}$$

или

$$R_{m2}^{(rov)} = 1 - [1 - r(t)]^{m+1} [1 + (m+1)r(t)]. \quad (4.24)$$

Если для системы n последовательных элементов предусмотрен только один резервный элемент, то используя выражение (4.22) при $m=1$ получим:

$$R_{1n}^{(rov)} = \sum_{i=n}^{n+1} C_i^{n+1} r^i(t) [1 - r(t)]^{n+1-i} = C_n^{n+1} r^n(t) [1 - r(t)] + r^{n+1}(t)$$

или

$$R_{1n}^{(rov)}(t) = r^n(t) + n[1 - r(t)]r^n(t). \quad (4.25)$$

4.8. Автономные устройства с общим резервированием

Для резервирования n независимых, автономных идентичных устройств может применяться общий резерв, представляющий собой m устройств, каждое из которых может заменить при отказе любое из n автономных устройств (рис. 4.2).

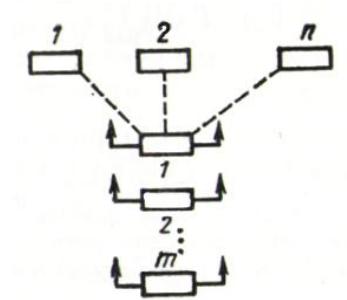


Рис. 4.2. Автономные устройства с общим резервированием

Если резервные устройства находятся в нагруженном режиме и переключатели работают безотказно, то надежность одного (любого) из автономных устройств зависит от наличия резервного компонента в случае отказа.

Обозначим через A_x событие, состоящее в том, что при отказе автономного устройства в момент времени x будет в наличии резервный компонент. Событие A_x произойдет в том случае, если при отказе автономного устрой-

ва среди остальных $m+n-1$ основных и резервных устройств отказало не более чем $m-1$ устройств. Тогда для замены отказавшего устройства найдется резервный компонент.

Вероятность безотказной работы каждого устройства (основного и резервного) равна $r(t)$. Тогда

$$R\{A_\tau\} = \sum_{i=0}^{m-1} C_i^{m+n-1} r^{m+n-1-i}(\tau) [1-r(\tau)]^i \quad (4.26)$$

Вероятность безотказной работы одного (любого) устройства при данном способе резервирования равна

$$R^{(n,m)}(t) = r(t) + \int_0^t w(\tau) R\{A_\tau\} r(t-\tau) d\tau, \quad (4.27)$$

где $w(\tau)$ – плотность вероятности момента отказа устройства.

Поскольку

$$R\{A_\tau\} r(t-\tau) \geq R\{A_\tau\}, \quad (4.28)$$

то формула (4.27) преобразуется к виду:

$$R^{(n,m)}(t) \geq r(t) + R\{A_\tau\} \int_0^t w(\tau) d\tau \quad (4.29)$$

или с учетом (4.26)

$$R^{(n,m)}(t) \geq r(t) + [1-r(t)] \sum_{i=0}^{m-1} C_i^{m+n-1} [r(t)]^{m+n-1-i} [1-r(t)]^i = R_0^{(n,m)}(t). \quad (4.30)$$

Формула (4.30) определяет нижнюю границу вероятности безотказной работы устройства.

4.9. Частные случаи

1) Имеется один резервный комплект для n основных. Из (4.27) и (4.30) при $m=1$ получаем

$$R^{(n,1)}(t) = r(t) + \int_0^t w(\tau) r^n(\tau) r(t-\tau) d\tau \geq r(t) + [1-r(t)] r^n(t). \quad (4.31)$$

2) На n автономных устройств имеется два резервных. Из (4.30) при $m=2$ получаем:

$$R_0^{(n,2)}(t) = r(t) + [1-r(t)] \{ r^{n+1}(t) + (n+1) r^n(t) [1-r(t)] \}. \quad (4.32)$$

4.10. Дублирование в ненагруженном режиме

Рассматривается простейшая система, состоящая из одного основного устройства и одного резервного в ненагруженном режиме. Вероятности безотказной работы от момента включения до первого отказа этих устройств равны соответственно

$$r_1(t) = \int_t^{\infty} w_1(x) dx, \quad (4.33)$$

$$r_2(t) = \int_t^{\infty} w_2(x) dx, \quad (4.34)$$

где $w_1(x), w_2(x)$ – плотности вероятности моментов отказа основного и резервного устройств.

Допустим, что переключающее устройство работает безотказно и действует мгновенно. Обозначим через ξ_1 момент отказа основного устройства и через $\tau_2 = \xi_1 + \xi_2$ момент отказа резервного устройства. Вероятность безотказной работы $R_2(t)$ рассматриваемой системы на интервале $(0, t)$ равна вероятности того, что $\tau_2 > t$:

$$R_2(t) = R\{\tau_2 > t\} = R\{\xi_1 + \xi_2 > t\} = \int_t^{\infty} W(x) dx, \quad (4.35)$$

где $W(x)$ – плотность вероятности момента τ_2 отказа системы.

Для определения плотности вероятности $W(x)$ разобьем интервал $(0, x)$ на n непересекающихся интервалов Δx и пусть x_k – точка, принадлежащая интервалу Δx_k , $k = 1, \dots, n$. Тогда вероятность того, что момент отказа системы будет находиться в интервале $(x, x + \Delta x)$, будет равна сумме произведений вероятности отказа основного устройства на интервале Δx_k на вероятность отказа резервного устройства на интервале $(x, x + \Delta x)$:

$$W(x)\Delta x = \sum_{k=1}^n w_1(x_k)\Delta x_k w_2(x - x_k)\Delta x.$$

При $\max_{1 \leq k \leq n} \Delta x_k \rightarrow 0$ получаем

$$W(x) = \int_0^x w_1(z) w_2(x - z) dz. \quad (4.36)$$

Подставляя (4.36) в (4.35), находим вероятность безотказной работы системы

$$R_2(t) = \int_0^t \int_0^{\infty} w_1(z) w_2(x-z) dz dx. \quad (4.37)$$

Изменение порядка интегрирования позволяет преобразовать выражение (4.37) так, как описано ниже. Область интегрирования на плоскости (x, z) находится выше заштрихованной линии на [рис. 2.4](#). Если интегрировать сначала по x , а затем по z , то интеграл разбивается на два, соответствующих подобластям I и II:

$$R_2(t) = \int_0^t \int_t^{\infty} w_1(z) w_2(x-z) dx dz + \int_t^{\infty} \int_0^{\infty} w_1(z) w_2(x-z) dx dz. \quad (4.38)$$

Замена $x - z = u$ приводит второй интеграл к виду

$$\int_t^{\infty} \int_0^{\infty} w_1(z) w_2(x-z) dx dz = \int_t^{\infty} \int_0^{\infty} w_1(z) w_2(u) du dz = \int_t^{\infty} w_1(z) dz = r_1(t), \quad (4.39)$$

где $r_1(t)$ – вероятность безотказной работы основного устройства.

Аналогичная замена в первом интеграле $x - z = u$ дает

$$\int_0^t \int_0^{\infty} w_1(z) w_2(x-z) dx dz = \int_0^t w_1(z) \int_{t-z}^{\infty} w_2(u) du dz = \int_0^t w_1(z) r_2(t-z) dz, \quad (4.40)$$

где $r_2(z)$ – вероятность безотказной работы резервного устройства.

Подставляя (4.39) и (4.40) в (4.38), получаем

$$R_2(t) = r_1(t) + \int_0^t w_1(z) r_2(t-z) dz. \quad (4.41)$$

Формула (4.41) имеет простую вероятностную интерпретацию. Безотказность на интервале $(0, t)$ рассматриваемой системы обеспечивается объединением двух событий: либо основное устройство работало безотказно на указанном интервале времени, либо основное устройство отказало в какой-то момент внутри этого интервала, но резервное, включившись в этот момент, работало безотказно на оставшемся отрезке времени до момента t .

Если учитывается ненадежность переключающего устройства и переключающее устройство может отказывать только в процессе переключения с постоянной вероятностью $f_{sw} = 1 - r_{sw}$, то в (4.41) функцию $r_2(t)$ следует заменить на $r_{sw} r_2(t)$. В этом случае вероятность безотказной работы рассматриваемой системы будет равна

$$R_2(t) = r_1(t) + r_{sw} \int_0^t w_1(z) r_2(t-z) dz. \quad (4.42)$$

4.11. Произвольное число резервных устройств в ненагруженном режиме

Рассматривается общий случай произвольного числа резервных устройств, находящихся в нагруженном режиме. Предположим, что переключающее устройство работает безотказно. Момент включения каждого устройства является случайной величиной, представляющей момент отказа предыдущего устройства (пренебрегаем временем, необходимым на переключение и на вхождение элемента в нагруженный режим).

Пусть число резервных устройств равно $m-1$. Обозначим через τ_i момент отказа i -го устройства, а через ζ_i интервал времени между $(i-1)$ -м и i -м отказом, $i = 2, 3, \dots, m$; $\zeta_1 = \tau_1$ – интервал безотказной работы основного устройства ($\tau_0 = 0$). Ясно, что

$$\tau_i = \sum_{k=1}^i (\tau_k - \tau_{k-1}) = \sum_{k=1}^i \zeta_k. \quad (4.43)$$

Вероятность безотказной работы системы в этом случае равна

$$R_m(t) = \int_t^{\infty} w_m(x) dx, \quad (4.44)$$

где $w_m(x)$ – плотность вероятности случайной величины τ_m .

Задача состоит в определении плотности суммы независимых случайных величин. Для этого следует использовать метод характеристических функций, имея в виду, что характеристическая функция суммы независимых случайных величин равна произведению характеристических функций слагаемых. Введем характеристическую функцию $\Theta_k(iv)$ интервала безотказной работы ζ_k -го резервного устройства. Тогда характеристическая функция случайной величины τ_m равна

$$\Theta_{\tau_m}(iv) = \prod_{k=1}^m \Theta_k(iv). \quad (4.45)$$

В случае, когда все устройства идентичные, получим

$$\Theta_{\tau_m}(iv) = \Theta^m(iv), \quad (4.46)$$

где $\Theta(iv)$ – характеристическая функция интервала безотказной работы любого устройства.

Подставляя в (4.44) вместо $w_m(x)$ ее выражение через характеристическую функцию, получим

$$w_m(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Theta_{\tau_m}(iv) e^{-ivx} dv;$$

запишем формулу вероятности безотказной работы в виде

$$R_m(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \prod_{k=1}^m \Theta_k(iv) e^{-ivx} dv dx, \quad (4.47)$$

и для идентичных устройств

$$R_m(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Theta^m(iv) e^{-ivx} dv dx. \quad (4.48)$$

Используя формулу (4.41), можно вероятность безотказной работы системы в рассматриваемом случае представить иначе:

$$R_m(t) = r_1(t) + \int_0^t w_1(z) r_m(t-z) dz, \quad m \geq 2, \quad (4.49)$$

где

$$r_m(x) = \int_{-\infty}^{\infty} \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} \prod_{k=1}^{m-1} \Theta_{k+1}(iv) e^{-ivx} dv \right] dx. \quad (4.50)$$

Если предположить, что переключающее устройство отказывает только в процессе переключения с вероятностью $f_{sw} = 1 - r_{sw}$, то из (4.49) находим вероятность безотказной работы рассматриваемой системы с учетом ненадежности переключающего устройства:

$$R_m(t) = r_1(t) + r_{sw}^{m-1} \int_0^t w_1(z) r_m(t-z) dz. \quad (4.51)$$

4.12. Резервирование системы при экспоненциальном распределении интервалов безотказной работы ее элементов

4.12.1. Система последовательных элементов. Экспоненциальный закон распределения, является наиболее распространенной на практике математической моделью безотказности элементов МЭМС в период нормальной эксплуатации [4]. Вероятность безотказной работе на любом интервале времени t в этом случае равна [21]

$$r(t) = e^{-\lambda t}, \quad t \geq 0, \quad \lambda > 0, \quad (4.52)$$

где λ – постоянная интенсивность отказов, равная величине, обратной среднему времени T_{50} безотказной работы.

Вероятность безотказной работы системы n последовательно соединенных элементов, отказы которых независимы и подчиняются экспоненциальному закону, равна произведению вероятностей безотказной работы ее элементов:

$$R_{1n}(t) = \prod_{j=1}^n \exp(-\lambda_j t) = \exp\left[-t \sum_{j=1}^n \lambda_j\right], \quad (4.53)$$

где λ_j - интенсивность отказов j -го элемента системы.

Из (4.53) следует, что для экспоненциального распределения длительности безотказной работы каждого элемента вероятность безотказной работы системы последовательно соединенных элементов также изменяется по экспоненциальному закону с интенсивностью Λ , равной сумме интенсивностей отказов элементов

$$\Lambda = \sum_{j=1}^n \lambda_j. \quad (4.54)$$

Среднее время безотказной работы T_{1n} системы в этом случае равно

$$T_{1n} = \frac{1}{\Lambda} = \left(\sum_{j=1}^n \frac{1}{T_{50j}} \right)^{-1}, \quad (4.55)$$

где T_{50j} – среднее время безотказной работы j -го элемента.

Если $\Lambda t \ll 1$, то вместо (4.53) можно пользоваться приближенной формулой

$$R_{1n}(t) \approx 1 - \Lambda t = 1 - t \sum_{j=1}^n \lambda_j.$$

Погрешность при этом не будет превышать величины $(\Lambda t)^2 / 2$.

Дисперсия времени безотказной работы для рассматриваемой системы равна

$$\sigma_{\tau}^2 = 1 / \Lambda^2. \quad (4.56)$$

Для однотипных элементов при $\lambda_j = \lambda$, $T_{50j} = T_{50}$ из (4.56) и (4.55) следует, что

$$\Lambda = n\lambda, \quad T_{1n} = T_{50} / n, \quad (4.57)$$

интенсивность отказов системы в n раз больше интенсивности отказов одного элемента, а среднее время безотказной работы системы в n раз меньше среднего времени безотказной работы одного элемента.

Вероятность безотказной работы рассматриваемой системы на интервале t , следующем за интервалом τ безотказной работы, равна

$$R_{1n}(t | \tau) = R_{1n}(t) = e^{-\lambda t}, \quad (4.58)$$

что соответствует экспоненциальному распределению.

4.12.2. Система параллельных элементов. При экспоненциальной модели безотказности элементов вероятность безотказной работы системы m параллельных элементов определяется выражением:

$$R_{m1}(t) = 1 - \prod_{i=1}^m (1 - e^{-\lambda_i t}), \quad (4.59)$$

Для идентичных элементов, когда $\lambda_j = \lambda$,

$$R_{m1}(t) = 1 - (1 - e^{-\lambda t})^m. \quad (4.60)$$

Если $\lambda t \ll 1$, то вместо (4.60) можно пользоваться приближенной формулой

$$R_{m1}(t) \approx 1 - (\lambda t)^m. \quad (4.61)$$

При этом погрешность не будет превышать величины $\frac{m}{2}(\lambda t)^{m+1}$.

В отличие от последовательного соединения в рассматриваемом случае вероятность безотказной работы не подчиняется экспоненциальному закону. Например, для системы двух однотипных параллельных элементов с интенсивностью отказов λ из (4.60) при $m = 2$ находим

$$R_{21}(t) = 1 - (1 - e^{-\lambda t})^2 = 2e^{-\lambda t} - e^{-2\lambda t}. \quad (4.62)$$

Условия вероятность безотказной работы на интервале t , следующем за интервалом τ безотказной работы, для системы m однотипных параллельных элементов определяется выражением

$$R_{m1}(t | \tau) = \frac{R_{m1}(t + \tau)}{R_{m1}(\tau)} = \frac{1 - [1 - e^{-\lambda(t+\tau)}]^m}{1 - (1 - e^{-\lambda\tau})^m}. \quad (4.63)$$

Интенсивность отказов рассматриваемой системы:

$$\lambda_{m1}(\tau) = \lim_{t \rightarrow 0} \frac{1 - R_{m1}(t | \tau)}{t}.$$

Используя (4.63) и переходя к пределу $t \rightarrow 0$, получим:

$$\lambda_{m1}(\tau) = m\lambda \frac{e^{-\lambda\tau} (1 - e^{-\lambda\tau})^{m-1}}{1 - (1 - e^{-\lambda\tau})^m}. \quad (4.64)$$

При $\tau = 0$ $\lambda_{m1}(0) = 0$; $\lim_{\tau \rightarrow \infty} \lambda_{m1}(\tau) = \lambda$.

Таким образом, интенсивность отказов системы однотипных параллельных элементов монотонно возрастает от нуля до величины λ , соответствующей интенсивности отказов одного элемента. Предельное значение $\lambda_{m1}(\infty) = \lambda$ не зависит от числа m параллельных элементов. При конечном значении τ интенсивность отказов системы тем меньше, чем больше m :

$$\lambda_{m_i,1}(\tau) < \lambda_{m_j,1}(\tau), \quad m_i > m_j.$$

Из (4.64) также следует, что при $m \rightarrow \infty$ (неограниченный комплект резервных элементов) предельное значение $\lambda_{\infty}(\tau) = 0$ при любом τ . При дублировании ($m=2$) из (4.64) следует

$$\lambda_{2,1}(\tau) = 2\lambda \frac{(1 - e^{-\lambda\tau})}{(2 - e^{-\lambda\tau})}.$$

4.12.3. Среднее время безотказной работы системы параллельных элементов. Из (4.59) находим среднее время безотказной работы системы m параллельных элементов

$$\begin{aligned} T_{m1} &= \int_0^{\infty} [1 - \prod_{i=1}^m (1 - e^{-\lambda_i t})] dt = \\ &= \sum_{j=1}^m (-1)^{j-1} \sum_{i_1=1}^{m-j-1} \sum_{i_2=i_1+1}^{m-j} \dots \sum_{i_j=i_{j-1}+1}^m (\lambda_{i_1} + \lambda_{i_2} + \dots + \lambda_{i_j})^{-1}, \quad i_0 = 0. \end{aligned} \quad (4.65)$$

Для однотипных элементов из (4.65) при $\lambda_i = \lambda$ получаем

$$T_{m1} = T_{50} \sum_{k=1}^m \frac{1}{k}. \quad (4.66)$$

Дисперсия времени безотказной работы в этом случае равна

$$\begin{aligned} \sigma_{Tm}^2 &= 2 \int_0^{\infty} t [1 - (1 - e^{-\lambda t})]^m dt - T_{m1}^2 = \\ &= \frac{2}{\lambda^2} \sum_{k=1}^m (-1)^{k+1} (C_k^m) \frac{1}{k^2} - \frac{1}{\lambda^2} \left(\sum_{k=1}^m \frac{1}{k} \right)^2 = \frac{1}{\lambda^2} \sum_{k=1}^m \frac{1}{k^2}. \end{aligned} \quad (4.67)$$

При $m \gg 1$

$$\sigma_{Tm}^2 \approx (\pi^2/6) T_{50}^2.$$

Из (4.67) следует, что среднее время безотказной работы системы m параллельных однотипных элементов больше среднего времени безотказной работы одного элемента в z_m раз, где величина z_m определяется по формуле:

$$z_m = \sum_{k=1}^m \frac{1}{k} \quad (4.68)$$

и представляет сумму первых m обратных чисел натурального ряда.

Из (4.66) и (4.67) следует, что отношение среднеквадратического значения к среднему значению времени безотказной работы убывает при $m \rightarrow \infty$ обратно пропорционально $\ln m$.

4.12.4. Раздельное дублирование системы. Рассмотрим систему n однотипных элементов, постоянное дублирование которой происходит r группами. Если λ – интенсивность отказов любого элемента, то из (2.15) при $m = 2$ находим

$$R_{2n}^{(r)}(t) = [1 - (1 - e^{-\lambda n t / r})^2]^r \quad (4.69)$$

и, следовательно,

$$\begin{aligned} T_{2n}^{(r)} &= \int_0^\infty [1 - (1 - e^{-\lambda n t / r})^2]^r dt = \sum_{k=1}^r \sum_{j=1}^{2k} (-1)^{k+j} (C_k^r) (C_j^{2k}) \int_0^\infty e^{-n j \lambda t / r} dt = \\ &= \frac{r}{n} T_{50} \sum_{k=1}^r \sum_{j=1}^{2k} (-1)^{k+j} (C_k^r) (C_j^{2k}) \frac{1}{j}. \end{aligned}$$

После суммирования получаем

$$T_{2n}^{(r)} = \frac{T_{50}}{n} \left[\frac{1}{2} + \frac{2^{2r-2} (r-1)!}{(2r-1)!} \right]. \quad (4.70)$$

Величина в квадратный скобках дает выигрыш от дублирования, так как у нерезервированной системы время безотказной работы в n раз меньше, чем среднее время каждого из элементов. Эта величина

$$z_{2n}^{(r)} = \frac{1}{2} + \frac{2^{2r-2} (r-1)!}{(2r-1)!} \quad (4.71)$$

зависит только от числа групп r и не зависит от n .

Предельные случаи:

1. При общем дублировании ($r = 1$)

$$z_{2n}^{(1)} = \frac{1}{2} + 1 = \frac{3}{2}; \quad T_{2n}^{(1)} = \frac{3}{2} \frac{T_{50}}{n}, \quad (4.72)$$

что согласуется с (2.82).

Абсолютное значение среднего времени зависит от числа элементов в системе, но выигрыш не зависит от числа элементов и равен $3/2$.

2. При дублировании по элементам ($r = n$)

$$z_{2n}^{(n)} = \frac{1}{2} + \frac{2^{2n-2} (n-1)!}{(2n-1)!}. \quad (4.73)$$

Используя асимптотическую формулу Стирлинга

$$n! \sim \sqrt{2\pi n} n^n e^{-n},$$

получаем при $n \gg 1$

$$z_{2n}^{(n)} \sim \frac{1}{2} + 0,885\sqrt{n}. \quad (4.74)$$

Существенным отличием от общего дублирования является то, что при дублировании по элементам эффективность резервирования растет с увеличением числа элементов.

4.12.5. Дублирование с учетом надежности переключающего устройства. Рассмотрим случай однократного нагруженного резервирования одного элемента с замещением. Обозначим через λ интенсивность отказов и основного, и резервного элемента, а через λ_a – интенсивность отказов переключающего устройства. Вероятность безотказной работы системы в рассматриваемом случае равна

$$R_{21}(t) = 1 - (1 - e^{-\lambda t})[1 - e^{-(\lambda + \lambda_a)t}] = e^{-\lambda t} + e^{-(\lambda + \lambda_a)t} - e^{-(2\lambda + \lambda_a)t} \quad (4.75)$$

Вводя обозначение

$$\beta = \lambda_a / \lambda \quad (4.76)$$

и используя (4.75), находим среднее время безотказной работы рассматриваемой дублированной системы

$$T_{21} = (1 + \frac{1}{1 + \beta} - \frac{1}{2 + \beta})T_{50} = [1 + 1/(1 + \beta)(2 + \beta)]T_{50} \quad (4.77)$$

Рассмотрим систему, состоящую из двух элементов, и найдем среднее время безотказной работы системы при дублировании в целом и по элементам с учетом ненадежности переключающего устройства. Вероятность безотказной работы при общем дублировании равна

$$R_{22}^1(t) = 1 - (1 - e^{-2\lambda t})[1 - e^{-(2\lambda + \lambda_a)t}], \quad (4.78)$$

а при раздельном дублировании

$$R_{22}^2(t) = [1 - (1 - e^{-\lambda t})(1 - e^{-(\lambda + \lambda_a)t})]^2. \quad (4.79)$$

Среднее время безотказной работы равно соответственно при общем дублировании

$$T_{22}^1 = \left[1 + \frac{1}{(1 + \beta/2)(2 + \beta/2)} \right] \frac{T_{50}}{2}, \quad (4.80)$$

при раздельном дублировании

$$T_{22}^2 = \left(1 + \frac{1}{(1 + \beta)} + \frac{1}{(2 + \beta)} + \frac{2}{(1 + \beta/2)} - \frac{2}{(3/2 + \beta/2)} - \frac{2}{(3/2 + \beta)} \right) \frac{T_{50}}{2}. \quad (4.81)$$

4.12.6. Скользящее резервирование. Рассмотрим систему однотипных последовательных элементов, для которой предусмотрено скользящее резервирование m элементами, находящимися в нагруженном режиме. Предполагается, что переключающие устройства работают безотказно. Из (4.23) для экспоненциального закона распределения интервалов безотказной работы основных и резервных элементов получаем

$$\begin{aligned} R_{mm}^{(c)}(t) &= \sum_{i=n}^{n+m} (C_i^{m+n}) e^{-i\lambda t} (1 - e^{-\lambda t})^{m+n-i} = \\ &= \sum_{i=n}^{n+m} \sum_{k=0}^{m+n-i} (-1)^k (C_i^{m+n}) (C_k^{m+n-i}) e^{-(i+k)\lambda t} \end{aligned} \quad (4.82)$$

где λ – интенсивность отказов одного элемента. Если имеется только один резервный элемент ($m = 1$), то

$$R_{1n}^{(c)}(t) = e^{-(n+1)\lambda t} + (n+1)e^{-n\lambda t}(1 - e^{-\lambda t}) \quad (4.83)$$

Из (4.82) следует, что среднее время безотказной работы системы равно

$$T_{mm}^{(c)} = \frac{1}{\lambda} \sum_{i=n}^{n+m} \sum_{k=0}^{m+n-i} (-1)^k (C_i^{m+n}) (C_k^{m+n-i}) \frac{1}{i+k} = \frac{1}{\lambda} \sum_{i=0}^m \frac{1}{n+i} \quad (4.84)$$

Если имеется только один резервный элемент ($m = 1$), то

$$T_{1n}^{(c)} = \frac{2n+1}{n(n+1)} T_{50}, \quad (4.85)$$

Где T_{50} – среднее время безотказной работы одного элемента. Выигрыш в среднем времени в этом случае равен

$$z_{2n}^{(c)} = n T_{1n}^{(c)} / T_{50} = (2n+1)/(n+1). \quad (4.86)$$

В частном случае, при $n = 1$, получается $z_{11}^{(c)} = 3/2$. При $n \rightarrow \infty$ имеем $z_{1n}^{(c)} \rightarrow 2$, т. е. принципиально невозможно получить выигрыш больше чем в два раза.

4.12.7. Автономные устройства с общим резервом. Рассмотрим систему n независимых автономных идентичных устройств, располагающих общим резервом, который состоит из m таких же устройств. Предполагается, что интервал безотказной работы каждого нерезервированного устройства распределен по экспоненциальному закону с параметром λ . Используя (4.30), находим нижнюю границу вероятности безотказной работы одного (любого) из устройств системы с учетом резервирования:

$$R_0^{(n,m)}(t) = e^{-\lambda t} + (1 - e^{-\lambda t}) \times \sum_{i=0}^{m-1} (C_i^{m+n-1}) e^{-(m+n-1-i)\lambda t} (1 - e^{-\lambda t})^i \quad (4.87)$$

В частном случае, когда имеется одно устройство на два основных ($m=1$, $n=2$), из (2.99) следует

$$R_0^{(2,1)}(t) = e^{-\lambda t} + e^{-2\lambda t} - e^{-3\lambda t}. \quad (4.88)$$

Точное значение вероятности безотказной работы любого из двух основных устройств равно в этом случае [см. (4.32)]

$$\begin{aligned} R^{(2,1)}(t) &= e^{-\lambda t} + \int_0^t \lambda e^{-\lambda \tau} e^{-2\lambda \tau} e^{-\lambda(t-\tau)} d\tau = \\ &= e^{-\lambda t} + \frac{1}{2} e^{-\lambda t} (1 - e^{-2\lambda t}) = \frac{3}{2} e^{-\lambda t} - \frac{1}{2} e^{-3\lambda t}. \end{aligned} \quad (4.89)$$

4.12.8. Дублирование элемента в ненагруженном режиме. Рассмотрим дублирование элемента при ненагруженном режиме резервного элемента без учета ненадежности переключающего устройства. Если основной и резервный элемент имеют одинаковую интенсивность отказов λ , то вероятность безотказной работы системы в этом случае равна [см. формулу (4.41)]

$$R_2(t) = e^{-\lambda t} + \int_0^t \lambda e^{-\lambda z} e^{-\lambda(t-z)} dz = e^{-\lambda t} (1 + \lambda t) \quad (4.89)$$

Вероятность того, что отказ рассматриваемой системы произойдет на интервале $(\tau, \tau + t)$, если до момента τ не было отказов, равна согласно (4.89)

$$q_2(t | \tau) = 1 - (1 + \frac{\lambda t}{1 + \lambda \tau}) e^{-\lambda t}. \quad (4.90)$$

Интенсивность отказов рассматриваемой системы равна

$$\lambda_2(\tau) = \lim_{t \rightarrow 0} \frac{q_2(t | \tau)}{t} = \lambda \frac{\lambda \tau}{1 + \lambda \tau}. \quad (4.91)$$

При изменении τ от нуля до бесконечности интенсивность отказов монотонно возрастает от нуля до величины, равной интенсивности отказов одного элемента. Среднее время T_2 безотказной работы рассматриваемой системы с ненагруженным резервом определяется в соответствии с (4.89)

$$T_2 = \int_0^{\infty} (1 + \lambda t) e^{-\lambda t} dt = 2T_{50}, \quad (4.92)$$

где T_{50} – среднее время безотказной работы одного элемента.

Соотношение (4.92) получается и непосредственно. Величина среднего времени безотказной работы равна среднему значению суммы двух случайных величин: ξ_1 – момента отказа первого элемента и ζ_2 – длительности интервала до отказа второго элемента, т. е.

$$T_2 = m_1\{\xi_1 + \zeta_2\} = m_1\{\xi_1\} + m_1\{\zeta_2\} = 2T_{50}.$$

Таким образом, при дублировании в ненагруженном режиме среднее время безотказной работы удваивается, в то время как при нагруженном режиме оно равно $1,5 T_{50}$.

4.12.9. Ненагруженное кратное резервирование. Рассмотрим теперь общий случай, когда имеется не один, $m-1$ резервных элементов. Предполагается, что резервные элементы находятся в ненагруженном режиме, переключающее устройство работает безотказно. Для определения вероятности безотказной работы воспользуемся формулой (4.48), предполагая, что все элементы однотипны и характеризуются постоянной интенсивностью отказов λ . Так как характеристическая функция для экспоненциального распределения равна [21]

$$\theta(i\nu) = \lambda/(\lambda - i\nu), \quad (4.93)$$

то характеристическая функция случайного момента τ_m отказа m -го элемента равна

$$\theta(i\nu) = \theta^m(i\nu) = \lambda^m/(\lambda - i\nu)^m. \quad (4.94)$$

Плотность вероятности момента времени τ_m находим обратным преобразованием Фурье, используя теорему о вычетах [21]:

$$W_{\tau_m}(t) = \frac{\lambda^m}{2\pi} \int_{-\infty}^{\infty} \frac{e^{-i\nu t}}{(\lambda - i\nu)^m} d\nu = \lambda \frac{(\lambda t)^{m-1}}{(m-1)!} e^{-\lambda t}, \quad t > 0. \quad (4.95)$$

Вероятность безотказной работы рассматриваемой системы определяется по формуле:

$$R_m(t) = \frac{\lambda}{(m-1)!} \int_t^{\infty} (\lambda y)^{m-1} e^{-\lambda y} dy = \frac{\lambda}{(m-1)!} \int_{\lambda t}^{\infty} z^{m-1} e^{-z} dz \quad (4.96)$$

Или используя обозначения полной и неполной гамма-функций, получаем:

$$R_m(t) = 1 - \Gamma(m, \lambda t) / \Gamma(m). \quad (4.97)$$

Из (4.96) и 4.97) находим выражение для интенсивности отказов:

$$\lambda(t) = \lambda \frac{(\lambda t)^{m-1}}{(m-1)!} e^{-\lambda t} [1 - \Gamma(m, \lambda t) / \Gamma(m)]^{-1}, \quad (4.98)$$

и, представляя неполную гамма-функцию в виде ряда [21], получаем

$$\lambda(t) = \lambda \frac{(\lambda t)^{m-1}}{(m-1)!} \left[\sum_{k=0}^{m-1} \frac{(\lambda t)^k}{k!} \right]^{-1} \quad (4.99)$$

Формула (4.91) является частным случаем (4.99) при $m=2$. Как и в указанном частном случае, при произвольном m интенсивность отказов монотонно возрастает от $\lambda(0) = 0$ до $\lambda(\infty) = \lambda$.

Так как момент отказа m -го элемента равен сумме интервалов безотказной работы всех элементов [см. (4.43)], то среднее время безотказной работы системы при ненагруженном режиме резервных элементов всегда (при любой модели безотказности элемента) в m раз больше среднего времени безотказной работы одного элемента:

$$T_m = m T_{50}. \quad (4.100)$$

В более общем случае, когда средние времена безотказной работы элементов не равны

$$T_m = \sum T_{50i} \quad (4.101)$$

где T_{50i} – среднее время безотказной работы i -го элемента.

Дисперсия времени безотказной работы в рассматриваемом случае равна сумме дисперсий времени безотказной работы элементов

$$\sigma_T^2 = \sum_{i=1}^m \sigma_{T_i}^2 = \sum_{i=1}^m T_{50i}^2.$$

При однотипных элементах $\sigma_T^2 = m T_{50}^2$. Отношение среднеквадратичного к среднему равно при этом $1/\sqrt{m}$.

4.12.10. Ненагруженное дублирование системы последовательно соединенных элементов. Рассмотрим случай, когда имеется система из n последовательных однотипных элементов, которая разбивается на r групп по $s=n/r$ элементов в каждой и каждая из таких групп дублируется в ненагруженном режиме. Вероятность безотказной работы любой группы равна [см. (4.89)] $(1 + n\lambda t / r) \times \exp(-n\lambda t / r)$. Вероятность безотказной работы системы находится как произведение таких вероятностей для r групп

$$R^{(r)}(t) = \left(1 + \frac{n\lambda t}{r} \right)^r e^{-n\lambda t}. \quad (4.102)$$

Частный случай этой формулы при общем резервировании ($r=1$):

$$R^{(1)}(t) = (1 + n\lambda t) e^{-n\lambda t}. \quad (4.103)$$

При резервировании по элементам ($r=n$)

$$R^{(n)}(t) = (1 + \lambda t)^n e^{-n\lambda t}. \quad (4.104)$$

Для общего резервирования ($r=1$) среднее время безотказной работы будет равно удвоенному значению величины T_{50}/n (среднего времени безотказной работы нерезервированной системы):

$$T^{(1)} = 2/n\lambda = 2T_{50}/n. \quad (4.105)$$

Увеличение среднего времени безотказной работы системы за счет дублирования в этом случае не зависит от n и равно двум.

В случае поэлементного резервирования среднее время безотказной работы системы в соответствии с (4.104) равно

$$T^{(n)} = \int_0^{\infty} (1 + \lambda t)^n e^{-n\lambda t} dt = \frac{e^n}{\lambda n^{n+1}} \int_n^{\infty} z^n e^{-z} dz$$

или

$$T^{(n)} = \frac{e^n n!}{n^{n+1}} \left[1 - \frac{\Gamma(n+1, n)}{\Gamma(n+1)} \right] T_{50} \quad (4.106)$$

Увеличение среднего времени безотказной работы за счет резервирования по элементам равно

$$z = \frac{nT^{(n)}}{T_{50}} = \frac{e^n n!}{n^n} \left[1 - \frac{\Gamma(n+1, n)}{\Gamma(n+1)} \right]. \quad (4.107)$$

Если $n > 1$ то, используя формулу Стирлинга, получаем

$$z \sim \sqrt{\pi n / 2}. \quad (4.108)$$

С увеличением числа n элементов системы абсолютное значение среднего времени безотказной работы при общем резервировании уменьшается пропорционально $1/n$, а при резервировании по элементам пропорционально $1/\sqrt{n}$.

4.13. Резервирование системы при неэкспоненциальном распределении интервалов безотказной работы ее элементов

4.13.1. Распределение Вейбулла. Начнем с определения вероятности безотказной работы системы n последовательно соединенных элементов, отказы которых независимы и подчиняются закону распределения Вейбулла:

$$P_{1n}(t) = \prod_{j=1}^n \exp(-\lambda_j t^{\delta_j}) = \exp\{-\sum_{j=1}^n \lambda_j t^{\delta_j}\}, \quad (4.109)$$

где λ_j, δ_j — параметры распределения j -го элемента системы. В том случае, когда параметры формы всех элементов одинаковы, т.е. когда $\delta_j = \delta$ из (4.109) получаем

$$P_{1n}(t) = e^{-\Lambda t^{\delta}}, \quad (4.110)$$

где

$$\Lambda = \sum_{j=1}^n \lambda_j. \quad (4.111)$$

Таким образом, при постоянном параметре формы интервал безотказной работы системы основных «вейбулловских» элементов также подчиняется вейбулловскому распределению с тем же параметром формы и вторым параметром Λ , равным сумме λ -параметров элементов. Среднее время безотказной работы в этом случае определяется:

$$T_{cp} = \Lambda^{-1/\delta} \Gamma(1 + 1/\delta) = \left(\sum_{j=1}^n \lambda_j \right)^{-1/\delta} \Gamma(1 + 1/\delta). \quad (4.112)$$

Но так как среднее время безотказной работы j -го элемента равно

$$t_{cpj} = \lambda_j^{-1/\delta} \Gamma(1 + 1/\delta).$$

то из (4.112) следует

$$T_{cp} = \left[\sum_{j=1}^n t_{cpj}^{-\delta} \right]^{-1/\delta}. \quad (4.113)$$

Для системы однотипных элементов, когда $t_{cpj} = t_{cp}$, имеем

$$T_{cp} = t_{cp}/n^\delta. \quad (4.114)$$

Из (4.114) следует, что для «стареющих» элементов ($\delta > 1$) отношение среднего времени работы системы n последовательно соединенных однотипных элементов к среднему времени безотказной работы одного элемента убывает в зависимости от n быстрее, чем $1/n$. Например, для $\delta = 2$, что соответствует релеевскому распределению, $T_{cp}/t_{cp} = 1/n^2$.

Вероятность безотказной работы системы m параллельных элементов, отказы которых независимы и подчиняются закону Вейбулла равна

$$P_{m1}(t) = 1 - \prod_{i=1}^m \left(1 - e^{-\lambda_i t^{\delta_i}} \right). \quad (4.115)$$

В случае однотипных элементов

$$P_{m1}(t) = 1 - \left(1 - e^{-\lambda t^{\delta}} \right)^m; \quad (4.116)$$

$$\varpi_{m1}(t) = m\lambda\delta t^{\delta-1} \left(1 - e^{-\lambda t^{\delta}} \right)^{m-1} e^{-\lambda t^{\delta}}. \quad (4.117)$$

В этом случае нетрудно определить среднее время безотказной работы системы:

$$\begin{aligned} T_{cp} &= \int_0^{\infty} \left[1 - \left(1 - e^{-\lambda t^{\delta}} \right)^m \right] dt = \\ &= \lambda^{-1/\delta} \Gamma(1 + 1/\delta) \sum_{k=1}^m (-1)^{k+1} \binom{m}{k} k^{-\delta}. \end{aligned} \quad (4.118)$$

Заметим, что при $\delta = 1$ (экспоненциальное распределение) из (4.118) следует

$$\sum_{k=1}^m (-1)^{k+1} \binom{m}{k} \frac{1}{k} = \sum_{k=1}^m \frac{1}{k}.$$

Из (4.115) и (4.116) нетрудно найти выражение для интенсивности отказов:

$$\begin{aligned}\lambda_{m1}(t) &= \varpi_{m1}(t)/P_{m1}(t) = \\ &= m\lambda\delta t^{\delta-1} (1 - e^{-\lambda t^\delta})^{m-1} e^{-\lambda t^\delta} [1 - (1 - e^{-\lambda t^\delta})^m]^{-1}.\end{aligned}\quad (4.119)$$

Если $\delta > 1$, то интенсивность отказов монотонно возрастает от $\lambda_{m1}(0) = 0$ до $\lambda_{m1}(\infty) = \lambda\delta$.

Как и в случае экспоненциального распределения, при фиксированном конечном значении t интенсивность отказов $\lambda_m(t)$ тем меньше, чем больше число m параллельных элементов.

Рассмотрим систему n однотипных элементов с однократным постоянным резервированием. При общем резервировании вероятность безотказной работы системы равна

$$P_{2n}^{(1)}(t) = 1 - (1 - e^{-n\lambda t^\delta})^2, \quad (4.120)$$

а среднее время безотказной работы

$$\begin{aligned}T_{2n}^{(1)}(t) &= \int_0^\infty (2e^{-n\lambda t^\delta} - e^{-2n\lambda t^\delta}) dt = \\ &= (n\lambda)^{-1/\delta} \Gamma(1 + 1/\delta) (2 - 2^{-1/\delta}).\end{aligned}\quad (4.121)$$

При резервировании по элементам вероятность безотказной работы системы равна

$$P_{2n}^{(n)}(t) = [1 - (1 - e^{-\lambda t^\delta})^2]^n, \quad (4.122)$$

а среднее время безотказной работы

$$\begin{aligned}T_{2n}^{(n)}(t) &= \int_0^\infty [1 - (1 - e^{-\lambda t^\delta})^2]^n dt = \\ &= \sum_{k=1}^n \sum_{j=1}^{2k} (-1)^{k+j} \binom{n}{k} \binom{2k}{j} \int_0^\infty e^{-j\lambda t^\delta} dt = \\ &= \lambda^{-1/\delta} \Gamma(1 + 1/\delta) \sum_{k=1}^n \sum_{j=1}^{2k} (-1)^{k+j} \binom{n}{k} \binom{2k}{j} j^{-1/\delta}.\end{aligned}\quad (4.123)$$

4.13.2. Усеченное нормальное распределение. Вероятность безотказной работы n последовательно соединенных элементов, отказы которых независимы и подчиняются усеченному нормальному распределению, равна

$$P_{1n}(t) = \prod_{j=1}^n \frac{1 - F\left(\frac{t - T_j}{\sigma_j}\right)}{F\left(T_j/\sigma_j\right)}. \quad (4.124)$$

где T_j, σ_j — параметры распределения j -го элемента системы. Если система состоит из однотипных элементов, то

$$P_{1n}(t) = \left[\frac{1 - F\left(\frac{t - T}{\sigma}\right)}{F\left(T/\sigma\right)} \right]^n \quad (4.125)$$

и

$$\varpi_{1n}(t) = \frac{n}{\sigma\sqrt{2\pi}} \frac{\left[1 - F\left(\frac{t-T}{\sigma}\right)\right]^{n-1}}{F^n(T/\sigma)} e^{-(t-T)^2/2\sigma^2}. \quad (4.126)$$

Из (4.125) следует, что среднее время безотказной работы системы однотипных элементов равно

$$T_{cp} = \sigma \int_{-\infty}^{T/\sigma} \left[\frac{F(z)}{F(T/\sigma)} \right]^n dz. \quad (4.127)$$

Вероятность безотказной работы системы m параллельных элементов, отказы которых независимы и подчиняются усеченному нормальному распределению, равна

$$P_{m1}(t) = 1 - \prod_{j=1}^m \frac{F\left(\frac{t-T_j}{\sigma_j}\right) + F\left(\frac{T_j}{\sigma_j}\right) - 1}{F(T_j/\sigma_j)}. \quad (4.128)$$

В случае однотипных элементов

$$P_{m1}(t) = 1 - \frac{\left[F\left(\frac{t-T}{\sigma}\right) + F(T/\sigma) - 1\right]^m}{F^m(T/\sigma)}. \quad (4.129)$$

4.14. Оптимизация затрат

4.14.1. Формулировки задач. При проектировании систем с резервированием необходимо не только удовлетворить требования к заданным показателям надежности, но и выполнить эти требования при минимальных затратах. Эта задача может ставиться двояко: а) при минимальных затратах удовлетворить требования к заданным показателям надежности; б) при затратах, не превышающих заданной величины, максимизировать показатель надежности. Под затратами могут подразумеваться не только стоимость, но и, например, масса или габариты.

Рассмотрим систему n последовательно соединенных независимых элементов, которые отличаются друг от друга надежностью и стоимостью. Предположим, что на заданном интервале времени $(0, t)$ вероятность безотказной работы i -го элемента равна $p_i(t)$ и что для этого элемента имеется $m_i - 1$ резервных ($i=1, \dots, n$). Вероятность $P(t)$ безотказной работы резервированной системы на заданном интервале времени $(0, t)$ зависит от $2n$ величин $p(t) = [p_1(t), \dots, p_n(t)]$, $m = [m_1, \dots, m_n]$:

$$P(t) = P[p(t), m]. \quad (4.130)$$

Обозначим через c_i стоимость i -го элемента рассматриваемой системы. Стоимость резервированной системы равна

$$C(c, m) = mc = \sum_{i=1}^n m_i c_i. \quad (4.131)$$

где c — вектор стоимостей элементов

$$C = (c_1, \dots, c_n).$$

Задачи оптимального резервирования можно теперь сформулировать следующим образом.

Первая задача. При заданных $p(t)$ и c найти такой вектор $\mathbf{m}^*$, которому соответствует нижняя грань стоимости системы $C(c, \mathbf{m})$ в области значений $\mathbf{m}$, удовлетворяющих условию $P[p(t), \mathbf{m}] \geq P$.

Вторая задача. При заданных $p(t)$ и c найти такой вектор $\mathbf{m}^*$, которому соответствует верхняя грань вероятности безотказной работы системы $P[p(t), \mathbf{m}]$ в области значений $\mathbf{m}$, удовлетворяющих условию $C(c, \mathbf{m}) \leq C$.

Результаты решения указанных задач оптимального резервирования используются для рационального планирования объема запасных комплектов (ЗИП). В этом случае, составляя схему расчета надежности, можно объединить условно в один элемент все однотипные элементы, даже если они конструктивно разнесены.

Заметим также, что в качестве показателя надежности в сформулированных задачах могут быть использованы среднее время безотказной работы или коэффициент готовности, а вместо одного показателя стоимости — несколько таких показателей.

Для точного решения сформулированных задач используются методы динамического программирования. Мы ограничимся приближенными решениями, полученными методом неопределенных множителей Лагранжа и методом наискорейшего спуска. Для высоконадежных систем эти приближенные решения получаются достаточно точными и не связаны с использованием специальной вычислительной техники.

4.14.2. Решение первой задачи методом неопределенных множителей. Рассмотрим случай постоянного резервирования. Тогда вероятность безотказной работы резервированной системы равна 1

$$P(\mathbf{p}, \mathbf{m}) = \prod_{i=1}^n [1 - (1 - p_i)^{m_i}]. \quad (4.132)$$

Необходимо найти такой вектор $\mathbf{m}^* = (m^*_1, \dots, m^*_n)$, при котором стоимость системы

$$C(c, \mathbf{m}^*) = \sum_{i=1}^n m^*_i c_i \leq \sum_{i=1}^n m_i c_i. \quad (4.133)$$

для всех m_i , удовлетворяющих неравенству

$$P(\mathbf{p}, \mathbf{m}) = \prod_{i=1}^n [1 - (1 - p_i)^{m_i}] \geq P. \quad (4.134)$$

Воспользуемся методом неопределенных множителей Лагранжа для определения условного экстремума функции $C(c, \mathbf{m})$ переменных $m_1, \dots, m_n$, пренебрегая дискретностью этих переменных. При этом решение задачи не получается в целых числах, что приводит к необходимости производить на заключительном этапе округление (с некоторым запасом) или перебор целочисленных значений, близких к полученному решению.

¹ Предполагается, что интервал времени фиксирован. Поэтому аргумент t в функциях везде опущен.

Введем обозначения

$$1 - (1 - p_i)^{m_i} = P^{\alpha_i}, \quad (4.135)$$

т.е. введем новые переменные α_i вместо переменных m_i и запишем формулу для нижней границы вероятности безотказной работы резервированной системы в виде

$$P = \prod_{i=1}^n P^{\alpha_i} = P^{\sum_{i=1}^n \alpha_i} \quad (4.136)$$

или

$$\sum_{i=1}^n \alpha_i = 1, \quad (4.137)$$

причем $0 < \alpha_i < 1$, $i=1, \dots, n$.

Из (4.135) следует

$$m_i = \ln(1 - P^{\alpha_i}) / \ln(1 - p_i). \quad (4.138)$$

Задача теперь сводится к следующей: найти оптимальные значения чисел α_i которые минимизируют стоимость

$$\sum_{i=1}^n m_i c_i = \sum_{i=1}^n c_i \frac{\ln(1 - P^{\alpha_i})}{\ln(1 - p_i)} = f(\alpha_1, \dots, \alpha_n) \quad (4.139)$$

при условии (4.137). Для определения экстремума запишем функцию Лагранжа

$$F(\alpha_1, \dots, \alpha_n) = f(\alpha_1, \dots, \alpha_n) - \lambda(\sum_{i=1}^n \alpha_i - 1), \quad (4.140)$$

где λ — неопределенный множитель Лагранжа, и составим систему уравнений

$$\frac{\partial F(\alpha_1, \dots, \alpha_n)}{\partial \alpha_j} = 0, \quad j = 1, \dots, n. \quad (4.141)$$

Подставляя (4.139) и (4.140) в (4.141), получим систему уравнений относительно неизвестных α_j :

$$\frac{c_j}{\ln(1 - p_j)} \frac{P^{\alpha_j} \ln P}{1 - P^{\alpha_j}} = \lambda, \quad j = 1, \dots, n, \quad (4.142)$$

в которой множитель λ определяется из условия (4.137). Каждое из уравнений системы (4.142) содержит только одно неизвестное, которое находится элементарными алгебраическими преобразованиями:

$$\alpha_j = -\ln \left[1 + \frac{c_j \ln P}{\lambda \ln(1 - p_j)} \right], \quad j = 1, \dots, n. \quad (4.143)$$

Условие (4.137) запишется теперь следующим образом:

$$\prod_{j=1}^n \ln \left[1 + \frac{c_j \ln P}{\lambda \ln(1 - p_j)} \right] = (-1)^n, \quad (4.144)$$

откуда получить величину неопределенного множителя λ в явном виде не удается.

4.14.3. Приближенное решение для высоконадежных систем. Введем дополнительное условие, состоящее в том, что вероятность безотказной работы резервированной системы должна быть близка к единице, т.е.

$$P = 1 - Q, \quad Q \ll 1. \quad (4.145)$$

Условие (4.145) с практической точки зрения представляет наибольший интерес. Из этого условия следует

$$P^{\alpha_k} = (1 - Q)^{\alpha_k} = 1 - \alpha_k Q + o(Q), \quad (4.146)$$

$$\ln P = \ln(1 - Q) = -Q + o(Q). \quad (4.147)$$

Подставляя (4.146) в (4.142), получаем

$$c_j / [\alpha_j \ln(1 - p_j)] = \lambda,$$

откуда

$$\alpha_j = c_j / [\lambda \ln(1 - p_j)].$$

Неопределенный множитель находим из условия (4.137):

$$\frac{1}{\lambda} \sum_{j=1}^n \frac{c_j}{\ln(1 - p_j)} = 1.$$

Следовательно, оптимальные значения чисел α_j при условии (4.145) равны 2

$$\alpha_j^* = u_j / \sum_{i=1}^n u_i, \quad j = 1, \dots, n, \quad (4.148)$$

где

$$u_j = c_j / \ln(1 - p_j) \quad (4.149)$$

Подставляя (4.148) в (4.138) и учитывая (4.145), находим оптимальные числа элементов:

$$m_j^* = \frac{\ln \alpha_j^* + \ln(1 - P)}{\ln(1 - p_j)} = \frac{\ln(1 - P)}{\ln(1 - p_j)} - \frac{\ln \sum_{i=1}^n u_i - \ln u_j}{\ln(1 - p_j)} \quad (4.150)$$

Первое слагаемое в (4.149), зависящее только от P и p_j представляет число параллельных элементов j -го вида, обеспечивающих заданную надежность P изолированной системы параллельных элементов этого вида. Дополнительное (положительное) слагаемое, зависящее от числа n последовательных элементов и их стоимостей, необходимо для обеспечения заданной надежности системы последовательных элементов всех n видов.

² Нетрудно показать, что решение (4.148) соответствует минимуму затрат при заданной вероятности P безотказной работы

Рассмотрим частные случаи формулы (4.150). Пусть все элементы имеют одинаковую надежность ($p_j = p$), но различные стоимости. Тогда из (4.148) следует

$$\alpha_j^* = c_j / \sum_{i=1}^n c_i / C_0, \quad (4.151)$$

где C_0 — стоимость нерезервированной системы. Из (4.150) при (4.151) имеем

$$m_j^* = \ln(1 - P) / \ln(1 - p) + c_j / [C_0 \ln(1 - p)]. \quad (4.152)$$

Если стоимости всех элементов одинаковы ($c_j = c$), а надежности различны, то

$$\alpha_j^* = \left[\ln q_j \sum_{j=1}^n \frac{1}{\ln q_j} \right]^{-1}, \quad q_j = 1 - p_j; \quad (4.153)$$

$$m_j^* = \frac{\ln(1-P)}{\ln(1-p)} + \left[\ln^2 q_j \sum_{j=1}^n \frac{1}{\ln q_j} \right]^{-1}. \quad (4.154)$$

При ($p_j = p$) и ($c_j = c$) из (4.148) следует $\alpha_j^* = 1/n$, и

$$m_j^* = \ln(1 - P^{1/n}) / \ln(1 - p), \quad (4.155)$$

что совпадает с результатом для постоянного поэлементного резервирования [см. также (2.138)].

4.14.4. Решение второй задачи методом неопределенных множителей. Пусть теперь задано ограничение на $m^* = (m_1^*, \dots, m_n^*)$, при котором вероятность безотказной работы системы с постоянным резервированием

$$P(p, m^*) = \prod_{i=1}^n [1 - (1 - p_i)^{m_i^*}] \geq \prod_{i=1}^n [1 - (1 - p_i)^{m_i}] \quad (4.156)$$

для всех m_i , удовлетворяющих неравенству

$$\sum_{i=1}^n m_i c_i \leq C. \quad (4.157)$$

Так как логарифм — монотонная функция, то сформулированная задача равносильна определению условного экстремума функции $\ln P(p, m)$ переменных $m_1, \dots, m_n$ при условии

$$\sum_{i=1}^n m_i c_i = C. \quad (4.158)$$

Учитывая, что

$$\ln P(p, m) = \sum_{i=1}^n \ln[1 - (1 - p_i)^{m_i}], \quad (4.159)$$

запишем функцию Лагранжа

$$F(m_1, \dots, m_n) = \sum_{i=1}^n \ln[1 - (1 - p_i)^{m_i}] + \lambda \sum_{i=1}^n m_i c_i \quad (4.160)$$

и составим систему уравнений

$$\frac{\partial F(m_1, \dots, m_n)}{\partial m_j} = 0, \quad j = 1, \dots, n, \quad (4.161)$$

предполагая, что аргументы функции $F(m_1, \dots, m_n)$ представляют любые действительные (положительные) числа.

Подставив (4.160) в (4.161), получим систему уравнений относительно неизвестных m_j

$$\frac{(1-p_j)^{m_j}}{1-(1-p_j)^{m_j}} \ln(1-p_j) + \lambda c_j = 0, \quad (4.162)$$

откуда находим

$$m_j = \frac{1}{\ln(1-p_j)} \ln \frac{\lambda c_j}{\lambda c_j - \ln(1-p_j)}, j = 1, \dots, n. \quad (4.163)$$

Условие (2.158) запишется теперь в виде

$$\sum_{j=1}^n c_j \frac{\ln \lambda c_j - \ln[\lambda c_j - \ln(1-p_j)]}{\ln(1-p_j)} = C. \quad (4.164)$$

Формула (4.164) позволяет определить неопределенный множитель λ в зависимости от заданных величин $p_1, \dots, p_n, c_1, \dots, c_n$.

4.14.5. Приближенное решение для высоконадежных систем. В этом случае в (4.159) логарифмы можно заменить их линейным членом разложения в ряд Тейлора, т.е.

$$\ln P(p, m) \approx -\sum_{i=1}^n (1-p_i)^{m_i} \quad (4.165)$$

и тогда вместо (4.162) получаем более простое уравнение

$$(1-p_j)^{m_j} + \frac{\lambda c_j}{\ln(1-p_j)} = 0, j = 1, \dots, n. \quad (4.166)$$

Используя обозначение $u_j = -c_j / \ln(1-p_j)$, совпадающее с точностью до знака с (4.149), запишем решение уравнения (4.166) в виде

$$m_j = \ln(\lambda u_j) / \ln(1-p_j). \quad (4.167)$$

Подставляя (4.167) в (4.157), находим

$$\ln \lambda \sum_{i=1}^n \frac{c_i}{\ln(1-p_i)} + \sum_{i=1}^n \frac{c_i \ln u_i}{\ln(1-p_i)} = C,$$

откуда получаем выражения для $\ln \lambda$:

$$\ln \lambda = -(C + \sum_{i=1}^n u_i \ln u_i) / \sum_{i=1}^n u_i. \quad (4.168)$$

Подставляя (4.168) в (4.167), находим оптимальные числа элементов каждого типа³

$$m_j^* = [\ln u_j - (C + \sum_{i=1}^n u_i \ln u_i) / \sum_{i=1}^n u_i] / \ln(1-p_j), j = 1, \dots, n. \quad (4.169)$$

³ Нетрудно показать, что решение (2.174) соответствует максимуму вероятности безотказной работы при заданной стоимости.

Рассмотрим частные случаи формулы (4.169). Если все элементы имеют одинаковую надежность $p_j=p$, но различные стоимости, то

$$m_j^* = \frac{C}{C_0} - \frac{1}{C_0 \ln(1-p)} \left(\sum_{i=1}^n c_i \ln \frac{c_i}{\ln(1-p)} - C_0 \ln \frac{c_j}{\ln(1-p)} \right), \quad (4.170)$$

где

$$C_0 = \sum_{i=1}^n c_i \quad (4.171)$$

— стоимость нерезервированной системы.

Если и надежности, и стоимости всех элементов одинаковы, то полагая в (4.170) $c_j = c$, получаем

$$m_j^* = m = C/C_0. \quad (4.172)$$

4.14.6. Метод наискорейшего спуска (подъема). Возможность получения решения рассмотренных выше задач в замкнутой форме методом неопределенных множителей Лагранжа была достигнута ценою ряда упрощающих предположений. Допускалось, что можно пренебречь дискретностью переменных $m_1, \dots, m_n$, что ограничения в виде неравенств можно заменить точными равенствами. В результате приходилось округлять полученные значения до ближайшего целого числа. Конечно, округление не всегда сопряжено с малыми погрешностями. Был рассмотрен, кроме того, только случай постоянного резервирования, который не охватывает задачи планирования объема запасных комплектов устройств, не находящихся в нагруженном режиме, причем использовалось требование высокой надежности резервированной системы. Более широкий класс задач оптимального резервирования в целочисленном виде можно решить так называемым методом наискорейшего спуска (подъема).

Метод наискорейшего спуска (подъема) является одним из эффективных итерационных методов отыскания минимума (максимума) функций многих переменных. Существо этого метода очень образно охарактеризовал Д. Уайльд в книге «Методы поиска экстремума» (Пер. с англ., М., «Наука», 1967, с.153): «Рассмотрим довольно частную задачу поиска вершины холма, сильно заросшего лесом и расположенного на низменности. Хотя из-за лесной чащи исследователь не только не видит вершины, но даже не может узнать холма, он в конце концов достигнет цели (в случае унимодальности) просто за счет того, что он непрерывно поднимается вверх. К вершине его приведет любая дорога, но если он поторопится, то, вероятно, он будет двигаться по тем направлениям, где наклон холма самый большой, при условии, конечно, что ему не придется при этом взбираться на отвесные скалы. Эта интуитивно привлекательная идея восхождения по кратчайшей дороге является основой метода поиска, известного под названием метода градиента или метода наискорейшего подъема».

Решение методом наискорейшего спуска (подъема) задач приводится к следующему многошаговому процессу вычислений⁴. На каждом шаге отыскивается такой участок резервируемой системы, прибавление к которому одного резервного элемента дает наибольшее увеличение вероятности безотказной работы системы в целом на единицу затрат.

Обозначим через $P_i(m_i)$ вероятность безотказной работы за фиксированное время i -го участка системы при наличии $m_i - 1$ резервных элементов, а через c_i , как и прежде, стоимость i -го элемента. Для каждого участка системы определяются $P_i(m_i)$, $m_i = 1, 2, \dots; i = 1, \dots, n$, и затем вычисляются величины

$$g_i(m_i) = [P_i(m_i + 1) - P_i(m_i)]/c_i P_i(m_i). \quad (4.173)$$

На первом шаге определяется такой участок с номером k , для которого $g_k(1) = \max_{1 \leq i \leq n} g_i(1)$ максимальна. К этому k -му участку добавляется один элемент и вычисляются значения

$$P^{(1)} = P^{(0)} P_k(2)/P_k(1); \quad (4.174)$$

$$C^{(1)} = C^{(0)} + c_k, \quad (4.175)$$

где $P^{(0)}$ и $C^{(0)}$ — вероятность безотказной работы и стоимость нерезервированной системы, равные

$$P^{(0)} = \prod_{i=1}^n P_i(1); C^{(0)} = \sum_{i=1}^n c_i. \quad (4.176)$$

На втором шаге определяется максимальная из величин $g_i(1), i \neq k, g_k(2)$. Добавляется один элемент к участку j , для которого

$$g_j(1) = \max_{1 \leq i \leq n, i \neq k} g_i(1), g_j(1) > g_k(2),$$

или снова к k -му участку, если $g_k(2) > g_j(1)$. Вычисляются значения

$$P^{(2)} = P_j(2)P^{(1)}/P_j(1), \text{ если } g_j(1) > g_k(2), \quad (4.177)$$

или

$$P^{(2)} = P_k(3)P^{(1)}/P_k(2), \text{ если } g_k(2) > g_j(1), \quad (4.178)$$

а также

$$C^{(2)} = C^{(1)} + c_j, \text{ если } g_j(1) > g_k(2), \quad (4.179)$$

или

$$C^{(2)} = C^{(1)} + c_k, \text{ если } g_k(2) > g_j(1). \quad (4.180)$$

Многошаговый процесс останавливается на таком шаге N , на котором выполняется условие

⁴ Доказательство приведено в статье Г. Блэка и Ф. Прошана, перевод которой включен в сборник [19], см. также [10 и 25].

$$p^{N-1} < P \leq p^N \quad (4.181)$$

при решении первой задачи, или условие

$$C^N < C \leq C^{N+1} \quad (4.182)$$

при решении второй задачи.

4.14.7 Пример. Проиллюстрируем методику оптимизации. Необходимо найти оптимальное число резервных элементов (при постоянном резервировании), обеспечивающее максимальную вероятность безотказной работы резервированной системы при условии, что стоимость резервированной системы не превышает 45 усл. ед. Результаты последовательных вычислений записываются в табл. 4.1.

Таблица 4.1

m_i			g_i			f	$p^{(i)}$	$c^{(i)}$
m_1	m_2	m_3	g_1	g_2	g_3			
1	1	1	0,5	0,1	0,02	0	0,315	9
2	1	1	0,167	0,1	0,02	1	0,473	10
3	1	1	0,0715	0,1	0,02	2	0,551	11
3	2	1	0,0715	0,023	0,02	3	0,716	14
4	2	1	0,0314	0,023	0,02	4	0,768	15
5	2	1	0,0162	0,023	0,02	5	0,792	16
5	3	1	0,0162	0,0065	0,02	6	0,840	19
5	3	2	0,0162	0,0065	0,0018	7	0,925	24
6	3	2	0,0078	0,0065	0,0018	8	0,940	25
7	3	2	0,0040	0,0065	0,0018	9	0,948	26
7	4	2	0,0040	0,0019	0,0018	10	0,967	29
8	4	2	0,0020	0,0019	0,0018	11	0,973	30
9	4	2	0,0010	0,0019	0,0018	12	0,978	31
9	5	2	0,0010	0,0013	0,0018	13	0,982	34
9	5	3	0,0010	0,0013	0,0002	14	0,988	39
9	6	3	0,0010	0,0002	0,0002	15	0,994	42
10	6	3	0,0005	0,0002	0,0002	16	0,995	43
11	6	3	0,0003	0,0002	0,0002	17	0,996	44
12	6	3	-	-	-	18	0,997	45

Находим величины $g_i(1)$ согласно (4.173):

$$g_1(1) = \frac{1 - (1 - p_1)^2 - p_1}{c_1 p_1} = \frac{1 - 0,5^2 - 0,5}{0,5} = 0,5$$

$$g_2(1) = \frac{1 - (1 - p_2)^2 - p_2}{c_2 p_2} = \frac{1 - 0,3^2 - 0,7}{3 \cdot 0,7} = 0,1$$

$$g_3(1) = \frac{1 - (1 - p_3)^2 - p_3}{c_3 p_3} = \frac{1 - 0,1^2 - 0,9}{5 \cdot 0,9} = 0,02$$

Удельное приращение вероятности безотказной работы – наибольшее для первого элемента. Поэтому добавляем к этому элементу один резервный элемент и вычисляем согласно (4.174) и (4.175)

$$P(1) = P(0) P_1(2)/P_1(1) = 0,315 \frac{(1-0,5^2)}{0,5} = 0,473$$

$$C(1) = C(0) + c_1 = 9+1=10.$$

Находим величину

$$g_1(2) = \frac{1-(1-p_1)^2-1+(1-p_1)^2}{c_1 [1-(1-p_1)^2]} = \frac{0,5^2-0,5^3}{1-0,5^2} = 0,167.$$

Так как $g_1(2) > g_2(1)$ и $g_1(2) > g_3(1)$, то снова добавляем к первому элементу один резервный элемент и вычисляем согласно (4.177) и (4.180)

$$P(2) = \frac{P_1(3)}{P_1(2)} P(1) = \frac{1-(1-p_1)^3}{1-(1-p_1)^2} P(1) = \frac{1-0,5^3}{1-0,5^2} 0,473 = 0,551,$$

$$C(2) = C(1) + c_1 = 10+1=11.$$

Находим величину

$$g_1(3) = \frac{(1-p_1)^3-(1-p_1)^4}{c_1 [1-(1-p_1)^3]} = \frac{0,5^3-0,5^4}{1-0,5^3} = 0,0715.$$

Так как $g_2(1) > g_1(3)$ и $g_2(1) > g_3(1)$, то снова добавляем ко второму элементу один резервный элемент и вычисляем

$$P(3) = \frac{P_2(2)}{P_2(1)} P(2) = \frac{1-(1-p_2)^2}{p_2} P(2) = \frac{1-0,3^2}{0,7} 0,551 = 0,716,$$

$$C(3) = C(2) + c_2 = 11+3=14.$$

Находим величину

$$g_2(2) = \frac{(1-p_2)^2-(1-p_2)^3}{c_2 [1-(1-p_2)^2]} = \frac{0,3^2-0,3^3}{3(1-0,3)^2} = 0,023.$$

Так как $g_1(3) > g_2(2)$, то добавляем к первому элементу еще один резервный элемент и вычисляем

$$P(4) = \frac{P_1(4)}{P_1(3)} P(3) = \frac{1-(1-p_1)^4}{1-(1-p_1)^3} P(3) = \frac{1-0,5^4}{1-0,5^3} 0,716 = 0,768,$$

$$C(4) = C(3) + c_1 = 14+1=15.$$

Находим величину

$$g_1(4) = \frac{(1-p_1)^4-(1-p_1)^5}{c_1 [1-(1-p_1)^4]} = \frac{0,5^4-0,5^5}{0,937} = 0,0314.$$

Так как $g_1(4) > g_2(2)$, то снова добавляем к первому элементу один резервный элемент и вычисляем

$$P(5) = \frac{P_1(5)}{P_1(4)} P(4) = \frac{1-(1-p_1)^5}{1-(1-p_1)^4} P(4) = \frac{1-0,5^5}{1-0,5^4} 0,768 = 0,792,$$

$$C(5) = C(4) + c_1 = 15+1=16.$$

Находим величину

$$g1(5) = \frac{(1-p_1)^5 - (1-p_1)^6}{c_1 [1 - (1-p_1)^5]} = \frac{0,5^5 - 0,5^6}{0,969} = 0,0162.$$

Так как $g2(2) > g1(5)$, то добавляем ко второму элементу один резервный элемент и вычисляем

$$P(6) = \frac{P_2(3)}{P_2(2)} P(5) = \frac{1 - (1-p_2)^3}{1 - (1-p_2)^2} P(5) = \frac{1 - 0,3^3}{1 - 0,3^2} 0,792 = 0,840,$$

$$C(6) = C(5) + c_2 = 16 + 3 = 19.$$

Находим величину

$$g2(3) = \frac{(1-p_2)^3 - (1-p_2)^4}{c_2 [1 - (1-p_2)^3]} = \frac{0,3^3 - 0,3^4}{3(1 - 0,3^3)} = 0,0065.$$

Так как $g3(1) > g1(5)$ и $g3(1) > g2(3)$, то добавляем к третьему элементу один резервный элемент и вычисляем

$$P(7) = \frac{P_3(2)}{P_3(1)} P(6) = \frac{1 - (1-p_2)^2}{p_3} P(6) = \frac{1 - 0,1^2}{0,9} 0,840 = 0,925,$$

$$C(7) = C(6) + c_3 = 19 + 5 = 24.$$

Дальнейшие вычисления аналогичны, они останавливаются, когда $c(12) = 45$. Результаты этих вычислений указаны в табл. 4.1. 7.

Многошаговый процесс вычислений, представленный в табл. 4.1 позволяет проследить увеличение вероятности безотказной работы и ее стоимости по мере роста кратности резервирования. Так, например, по этой таблице можно заключить, что для системы, для которой требуемая вероятность безотказной работы равна $P \geq 0,95$, число резервных элементов $m_1 - 1 = 6$, $m_2 - 1 = 3$, $m_3 - 1 = 1$ обеспечивает минимальную стоимость системы $c = 29$.

4.14.8. Пример расчета оптимального числа запасных элементов. В качестве второй иллюстрации метода наискорейшего спуска рассмотрим задачу определения оптимального числа запасных комплектов элементов каждого типа для описанной выше системы. Запасные элементы находятся в ненагруженном режиме. Поэтому необходимо задать в явном виде закон распределения интервалов безотказной работы элементов, находящихся в рабочем режиме. Предположим, что длительности безотказной работы элементов распределены экспоненциально с интенсивностями

$$\lambda_1 = 10^{-3} \text{ ч}^{-1}, \lambda_2 = 0,5 \cdot 10^{-3} \text{ ч}^{-1}, \lambda_3 = 10^{-4} \text{ ч}^{-1},$$

а стоимости их такие, как указано в п. 2.4.5, т.е. $c_1 = 1$, $c_2 = 3$, $c_3 = 5$.

Для периода эксплуатации $t = 1000$ ч требуется обеспечить вероятность безотказной работы системы $P(t) \geq 0,999$ при минимальных затратах.

Для определения величин $P_i(m_i)$ и $G_i(m_i)$ в многошаговой процедуре наискорейшего спуска следует воспользоваться формулой (4.97) для вероятности безотказной работы каждого участка системы при ненагруженном кратном

резервировании и экспоненциальном распределении длительности безотказной работы:

$$P_i(m_i) = 1 - \frac{\Gamma(mi, \lambda_i t)}{\Gamma(mi)}, \quad (4.183 \text{ } 2.186)$$

$$P_i(m_i + 1) - P_i(m_i) = \frac{(\lambda_i t)^{m_i}}{m_i!} e^{-\lambda_i t}. \quad (4.184 \text{ } 2.187)$$

Для указанных выше условий

$$\lambda_1 t = 1, \lambda_2 t = 0,5, \lambda_3 t = 0,1.$$

При вычислении величин $g_i(m_i)$ и $P^{(k)}$ следует воспользоваться для определения правой части в формулах (4.183) и (4.184) таблицами пуассоновского распределения.

Проведем последовательно вычисления, предписываемые методом наискорейшего спуска:

$$P_1(1) = e^{-\lambda_1 t} = 1/e = 0,368, P_2(1) = e^{-\lambda_2 t} = e^{-0,5} = 0,606, P_3(1) = e^{-\lambda_3 t} = e^{-0,1} = 0,906;$$

$$P(0) = e^{-(\lambda_1 + \lambda_2 + \lambda_3)t} = 0,368 * 0,606 * 0,906 = 0,202,$$

$$C(0) = c_1 + c_2 + c_3 = 1 + 3 + 5 = 9;$$

$$g_1(1) = \frac{0,3679}{0,3679} = 1, g_2(1) = \frac{0,3033}{3 * 0,6065} = 0,166,$$

$$g_3(1) = \frac{0,0905}{5 * 0,9048} = 0,020;$$

$$P(1) = 0,202 \frac{0,736}{0,368} = 0,405, C(1) = 10,$$

$$g_1(2) = 0,1839 / 0,7358 = 0,26,$$

$$P(2) = 0,405(0,920 / 0,736) = 0,507, C(2) = 11,$$

$$g_1(3) = 0,0613 / 0,920 = 0,067,$$

$$P(3) = 0,507 * 0,910 / 0,606 = 0,760, C(3) = 14,$$

$$g_2(2) = 0,0758 / 0,910 * 3 = 0,0278,$$

$$P(4) = 0,76 * 0,981 / 0,920 = 0,810, C(4) = 15,$$

$$g_1(4) = 0,0153 / 0,981 = 0,0156,$$

$$P(5) = 0,81 * 0,986 / 0,910 = 0,880, C(5) = 18,$$

$$g_2(3) = 0,0126 / 0,986 * 3 = 0,0042,$$

$$P(6) = 0,88 * 0,995 = 0,905, C(6) = 23,$$

$$g_3(2) = 0,00452 / 0,995 * 5 = 0,0009,$$

$$P(7) = 0,969 * 0,996 / 0,981 = 0,984, C(7) = 24,$$

$$g_1(5) = 0,0031 / 0,996 = 0,0032,$$

$P(8) = 0,984 \cdot 0,998 / 0,986 = 0,996$, $C(8) = 27$,
 $g_2(4) = 0,0016 / 0,998 \cdot 3 = 0,0005$,
 $P(9) = 0,996 \cdot 0,9994 / 0,0083 = 0,997$, $C(9) = 28$,
 $g_1(6) = 0,0005 / 0,9994 = 0,0005$,
 $P(10) = 0,997 \cdot 0,9998 / 0,9960 = 0,999$, $C(10) = 33$.
 Результаты расчетов сведены в табл. 4.2.

Таблица 4.2

m_i			g_i			f	$p^{(j)}$	$c^{(j)}$
m_1	m_2	m_3	g_1	g_2	g_3			
1	1	1	1	0,166	0,020	0	0,202	9
2	1	1	0,26	0,166	0,020	1	0,405	10
3	1	1	0,067	0,166	0,020	2	0,507	11
3	2	1	0,067	0,0278	0,020	3	0,760	14
4	2	1	0,0156	0,0278	0,020	4	0,810	15
4	3	1	0,0156	0,0042	0,020	5	0,880	18
4	3	2	0,0156	0,0042	0,0009	6	0,969	23
5	3	2	0,0032	0,0042	0,0009	7	0,984	24
5	4	2	0,0032	0,0005	0,0009	8	0,996	27
6	4	2	0,0005	0,0005	0,0009	9	0,997	28
6	4	3	0,0005	0,0005	0,0003	10	0,999	33

Таким образом, при оптимальном резервировании системы, когда резервные элементы находятся в ненагруженном режиме, за счет увеличения стоимости системы в $33/9 \approx 3,6$ раза вероятность ее отказа за время $t=1000$ ч уменьшается от 0,8 до 10^{-3} .

5. НАДЕЖНОСТЬ ВОССТАНАВЛИВАЕМЫХ СИСТЕМ

5.1. Элементы теории массового обслуживания

5.1.1. Предмет теории. Существует класс практически важных задач систем массового обслуживания, требующих применения вероятностных методов. Примерами таких систем являются системы многоканальной телефонной связи и некоторые системы автоматического управления, системы обслуживания одним рабочим или бригадой большого числа машин. К системам массового обслуживания относятся многие службы быта (парикмахерские, кассы, комбинаты обслуживания и т.д.); организация движения на железнодорожном, автомобильном, воздушном и других видах транспорта; сложная система энергоснабжения как в масштабе одного предприятия, так и в масштабе страны или группы стран, система обороны от нападения противника; система сбора и обработки информации и, наконец, система обеспечения надежности радиотехнических комплексов.

В терминах надежности поток требований на обслуживание – это поток отказов аппаратуры, а обслуживание требований – устранение отказов, т.е. восстановление работоспособности аппаратуры.

Задача системы массового обслуживания состоит в том, чтобы выполнить требования на обслуживание, поступающие в систему в случайные моменты времени. Естественно, в зависимости от интенсивности поступления требований, числа каналов и времени обслуживания, система будет обладать определенной эффективностью. В теории массового обслуживания исследуются количественные характеристики систем, позволяющие сравнивать между собой их эффективность.

5.1.2 Поток требований и время обслуживания. Последовательность случайных событий, состоящих в появлении требования на обслуживание, называют потоком требований. Обозначим через $x(t)$ число требований, поступающих в систему массового обслуживания на интервале времени $(0, t)$, т.е. от начала работы до какого-то момента времени t . Из-за случайной природы потока требований величина $x(t)$ – случайная, зависящая от параметра t .

Одной из основных моделей потока требований в теории массового обслуживания является модель простейшего потока, удовлетворяющего условиям стационарности, ординарности и отсутствия последствия. В дальнейшем будем предполагать, что поток требований – простейший.

В качестве распределения вероятностей времени обслуживания в теории массового обслуживания чаще всего используется экспоненциальное распределение. Это распределение и принято в дальнейшем изложении.

5.1.3. Процесс обслуживания как марковский случайный процесс. Рассмотрим систему массового обслуживания, состоящую из каналов. Пусть в начальный момент времени обслуживанием занято r каналов. В произвольный момент времени t число $N(t)$ каналов, занятых обслуживанием, – случайная

величина, которая зависит от того, сколько требований поступило в систему от начала отсчета времени до момента времени t и сколько требований за это время покинуло систему, будучи уже обслуженными.

Случайная величина $N(t)$, зависящая от текущего времени t , представляет случайный процесс. При сделанных предположениях о характере входного потока и времени обслуживания этот процесс обладает следующим свойством: если в некоторый момент времени t_0 число занятых обслуживанием каналов r т.е. $N(t_0)=r$, то значения $N(t)$ при $t > t_0$ не будут зависеть от того, что происходило в системе до момента времени t_0 . Это свойство является прямым следствием предположения о том, что поток требований – простейший, а распределение времени обслуживания – экспоненциальное. В любой момент времени число каналов $N(t)$, занятых обслуживанием, определяется тремя факторами: числом тех каналов из общего числа r обслуживаемых в момент t_0 , которые заканчивают обслуживание до момента времени t , числом новых требований, поступающих в систему, временем обслуживания вновь поступивших требований. Ни один из этих факторов не зависит от предыстории процесса.

Первый фактор – длительность обслуживания тех требований, которые уже обслуживались в момент времени t_0 , не зависит от того, сколько времени продолжалось обслуживание до момента t_0 , что характерно для экспоненциального распределения времени обслуживания.

Второй фактор – число новых требований, поступающих после t_0 , не зависит от того, сколько требований поступило до момента t_0 , так как поток требований – простейший.

Третий фактор – обслуживание требований, начатое после момента t_0 , не зависит от того, как происходило обслуживание в системе до момента t_0 .

Таким образом, случайный процесс $N(t)$, т.е. число каналов, занятых обслуживанием в момент времени полностью характеризуется условной вероятностью

$$P \{N(t) = k | N(t_0) = r\}, t > t_0.$$

Это означает, что процесс обслуживания при простейшем потоке требований и экспоненциальном распределении времени обслуживания является марковским случайным процессом.

5.1.4. Основные типы систем массового обслуживания. Классификация систем массового обслуживания основывается на различных принципах организации обслуживания.

Одним типом системы массового обслуживания является система, в которой приходящее требование получает отказ в обслуживании, если все каналы системы заняты. Это так называемая система с потерями, т.е. система, в которой требование теряется, если приходит в тот момент, когда все каналы заняты. В дальнейшем процесс обслуживания протекает так, как если бы вообще это требование не поступало. Характеристикой систем с потерями является вероятность того, что все каналы окажутся занятыми. Пропускную

способность такой системы характеризует среднее число загруженных каналов.

Другим типом системы массового обслуживания является система с очередями. В этой системе требование, пришедшее в систему, когда все ее каналы заняты, становится в очередь и ждет обслуживания. В системе с очередями требование не теряется, а ждет обслуживания. Характеристиками систем с ожиданием являются распределение вероятностей времени ожидания, среднее время ожидания.

Наконец, может существовать смешанная система (с очередью и с потерями). Разновидностью смешанной системы является такая, в которой поступающие требования становятся в очередь только в том случае, когда число ранее прибывших и ожидающих обслуживания требований не превосходит заданного числа m ; в противном случае требование теряется. В смешанных системах другой группы потеря требования происходит в том случае, когда время ожидания до обслуживания превосходит заданную величину τ . Характеристиками смешанных систем являются вероятность занятости всех каналов, среднее время ожидания до обслуживания.

При анализе систем массового обслуживания принимают обычно некоторые дополнительные предположения. Одним из них является предположение о том, что все свободные каналы равнодоступны всем приходящим требованиям. Иногда вводится предположение, что некоторые каналы закрепляют за привилегированными требованиями. Обслуживание может происходить в порядке поступления требований, в случайном порядке или с предпочтением некоторым требованиям (с приоритетами)

5.1.5. Переходные вероятности в системе с потерями. В такой системе до тех пор, пока не заняты все каналы, приходящее требование начинает обслуживаться немедленно. Если же требование поступает, когда все n каналов заняты, то оно теряется. Будем называть i -м состоянием системы с потерями такое, при котором i каналов системы заняты обслуживанием, $i=0, 1, \dots, n$. Потери могут возникать лишь тогда, когда система находится в n -м состоянии.

Обозначим через $p_i(t)$ вероятность того, что система в произвольный момент времени t находится в i -м состоянии. Последовательность функций $p_0(t), \dots, p_n(t)$ характеризует (в вероятностном смысле) систему массового обслуживания с потерями. Предположим, что поток требований – простейший, с параметром λ , а время обслуживания распределено по экспоненциальному закону с параметром μ . Найдем зависимости функций $p_k(t)$, $k=0, \dots, n$ от λ и μ . Для этого рассмотрим сначала два момента времени: t и $t+\Delta t$ и условную вероятность того, что система, находившаяся в момент времени t в i -м состоянии, будет в момент времени $t+\Delta t$ находится в k -м состоянии:

$$P_{ik}(\Delta t) = P\{N(t + \Delta t) = k | N(t) = i\} \quad (5.1)$$

Условную вероятность $P_{ik}(\Delta t)$ называют переходной вероятностью. При указанных предположениях относительно потока требований и времени обслуживания переходные вероятности зависят только от Δt . Ясно, что

$$\sum_{k=0}^n p_{ik}(\Delta t) = 1. \quad (5.2)$$

Напомним, что для простейшего потока

$$P_{>1}(\Delta t) = o(\Delta t).$$

При экспоненциальном времени обслуживания (с параметром μ) вероятность того, что одно требование будет обслуживаться не менее чем Δt , равна $e^{-\mu\Delta t}$. Если каналы независимы, то вероятность того, что $k \leq n$ каналов будут заняты обслуживанием не менее чем Δt , равна $e^{-k\mu\Delta t}$. Вероятность того, что из k работающих каналов освободится за время Δt хотя бы один, равна

$$1 - e^{-k\mu\Delta t} = k\mu\Delta t + o(\Delta t). \quad (5.3)$$

А вероятность освобождения $m > 1$ каналов равна

$$(1 - e^{-k\mu\Delta t})^m = o(\Delta t). \quad (5.3')$$

Теперь обратимся к переходным вероятностям $P_{ik}(\Delta t)$. Рассмотрим сначала случай перехода больше через одно состояние $|i - k| > 1$. При этом разность между числом прибывающих и покидающих требований по абсолютному значению больше или равна двум. Вероятность прихода двух и более требований — малая величина более высокого порядка, чем Δt . Вероятность окончания обслуживания двух и более требований есть тоже малая величина более высокого порядка, чем Δt . И, наконец, случай, когда или приходит более двух, или убывает более двух требований, есть также малая величина более высокого порядка, чем Δt . Итак, вероятность перехода через два или большее число состояний есть малая величина более высокого порядка, чем Δt :

$$p_{ik}(\Delta t) = o(\Delta t), \quad |i - k| > 1. \quad (5.4)$$

Вероятность перехода из состояния k в состояние $k+1$ равна

$$p_{k,k+1}(\Delta t) = \lambda\Delta t + o(\Delta t), \quad k < n. \quad (5.5)$$

Используя (3.3), находим вероятность перехода из состояния k в $k-1$

$$p_{k,k-1}(\Delta t) = k\mu\Delta t + o(\Delta t), \quad k > 0. \quad (5.6)$$

Рассматривая полную группу событий, получаем [см. (3.2) и рис. 3.1]

$$p_{k,k-1}(\Delta t) + p_{k,k}(\Delta t) + p_{k,k+1}(\Delta t) + p_{|k-i|>1}(\Delta t) = 1. \quad (5.7)$$

Подставляя в (5.7) выражения для переходных вероятностей из (5.4), (5.5) и (5.6), находим при $0 < k < n$

$$p_{kk}(\Delta t) = 1 - \lambda\Delta t - k\mu\Delta t + o(\Delta t) \quad (5.8)$$

Теперь рассмотрим граничные случаи $k = 0$ и $k = n$, которые отличаются от общего случая. При $k = 0$ $p_{k,k-1} = 0$, и из (5.7) находим

$$P_{00}(\Delta t) = 1 - \lambda \Delta t + o(\Delta t) \quad (5.9)$$

При $k = n$ $p_{n,n+1} = 0$, и из (5.7) находим

$$P_{nn}(\Delta t) = 1 - n\mu \Delta t + o(\Delta t) \quad (5.10)$$

5.1.6. Система дифференциальных уравнений для определения вероятностей состояний системы с потерями. Запишем вероятность того, что в момент времени $t + \Delta t$ система находится в нулевом состоянии (полностью не загружена), используя формулу полной вероятности $p_0(t + \Delta t) = p_0(t)p_{00}(\Delta t) + p_1(t)p_{10}(\Delta t) + \sum_{r=2}^n p_r(t)p_{r0}(\Delta t) = p_0(t)(1 - \lambda \Delta t) + p_1(t)\mu \Delta t + o(\Delta t)$, откуда при $\Delta t \rightarrow 0$

$$p'_0(t) = -\lambda p_0(t) + \mu p_1(t) \quad (5.11)$$

Аналогично при $0 < k < n$ получим, используя формулу полной вероятности и (5.4)-(5.7),

$$\begin{aligned} p_k(t + \Delta t) &= p_{k-1}(t)p_{k-1,k}(\Delta t) + p_k(t)p_{k,k}(\Delta t) + p_{k+1}(t)p_{k+1,k}(\Delta t) + o(\Delta t) \\ &= p_{k-1}(t)\lambda \Delta t + p_k(t)(1 - \lambda \Delta t - k\mu \Delta t) + p_{k+1}(t)(k+1)\mu \Delta t + o(\Delta t) \end{aligned}$$

Перенеся величину $p_k(t)$ в левую часть и разделив обе части равенства на Δt , получим после перехода к пределу $\Delta t \rightarrow 0$

$$\begin{aligned} p'_k(t) &= \lambda p_{k-1}(t) - (\lambda + k\mu)p_k(t) + (k+1)\mu p_{k+1}(t), \\ 0 &< k < n. \end{aligned} \quad (5.12)$$

Наконец, рассмотрим вероятность n -го состояния в момент $t + \Delta t$:

$$\begin{aligned} p_n(t + \Delta t) &= p_{n-1}(t)p_{n-1,n}(\Delta t) + p_n(t)p_{n,n}(\Delta t) + o(\Delta t) \\ &= \lambda \Delta t p_{n-1}(t) + (1 - n\mu \Delta t)p_n(t) + o(\Delta t), \end{aligned}$$

откуда при $\Delta t \rightarrow 0$ находим

$$p'_n(t) = \lambda p_{n-1}(t) - n\mu p_n(t) \quad (5.13)$$

Объединяя (5.11), (5.12) и (5.13), запишем систему дифференциальных уравнений относительно вероятностей состояний системы с потерями:

$$\left. \begin{aligned} p'_0(t) &= -\lambda p_0(t) + \mu p_1(t), \\ p'_k(t) &= \lambda p_{k-1}(t) - (\lambda + k\mu)p_k(t) + (k+1)\mu p_{k+1}(t), \quad 0 < k < n, \\ p'_n(t) &= \lambda p_{n-1}(t) - n\mu p_n(t). \end{aligned} \right\} \quad (5.14)$$

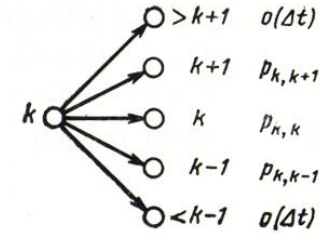


Рис 5.1 Граф перехода состояний системы массового обслуживания

Начальным условием для системы (5.14) является распределение вероятностей состояний $p_k(0)$, $k=0, 1, \dots, n$ в начальный момент времени $t=0$.

Совершая над обеими частями каждого из уравнений (5.14) преобразование Лапласа, получим систему линейных алгебраических уравнений. Если в начальный момент времени система не загружена т.е. $p_0(0)=1$, $p_k(0)=0, k>0$, то из (5.14) следует

$$\left. \begin{aligned} (s + \lambda)\Phi_0(s) - \mu\Phi_1(s) &= 1, \\ -\lambda\Phi_{k-1}(s) + (s + \lambda + k\mu)\Phi_k(s) - \\ -(k+1)\mu\Phi_{k+1}(s) &= 0, 0 < k < n; \\ -\lambda\Phi_{n-1}(s) + (s + n\mu)\Phi_n(s) &= 0, \end{aligned} \right\} \quad (5.15)$$

где

$$\Phi_k(s) = \int_0^\infty p_k(t)e^{-st} dt, k = 0, 1, \dots, n. \quad (5.15')$$

5.1.7. Предельные вероятности состояний системы с потерями. Для многоканальных систем массового обслуживания ($n \gg 1$) решение системы уравнений (5.14) связано с громоздкими вычислениями. Однако во многих случаях наибольший интерес представляют вероятности состояний через достаточно большое время после начала функционирования системы обслуживания. Иначе говоря, представляет интерес исследование асимптотического поведения функций $p_k(t)$ при $t \rightarrow \infty$. Это исследование основывается на теореме Маркова о существовании (для широкого класса марковских процессов) предельных вероятностей

$$\lim_{t \rightarrow \infty} p_k(t) = p_k. \quad (5.16)$$

Из (5.16) следует также

$$\lim_{t \rightarrow \infty} p'_k(t) = 0. \quad (5.16')$$

Тогда из (5.14) при $t \rightarrow \infty$ получаем систему линейных алгебраических уравнений относительно предельных вероятностей состояний:

$$\left. \begin{aligned} -\lambda p_0 + \mu p_1 &= 0, \\ \lambda p_{k-1} - (\lambda + k\mu)p_k + (k+1)\mu p_{k+1} &= 0, \\ \lambda p_{n-1} - n\mu p_n &= 0 \end{aligned} \right\} \quad (5.17)$$

при очевидном условии

$$\sum_{k=0}^n p_k = 1. \quad (5.18)$$

Обозначим

$$y_{k=\lambda} p_{k-1} - k\mu p_k. \quad (5.19)$$

Тогда из (5.17) при $0 < k < n$ следует, что $y_k - y_{k+1} = 0$. Но из первого уравнения (5.17) имеем $y_1 = 0$, а из последнего $y_1 = 0$. Отсюда следует, что $y_k = 0$ при любом k_i т.е. что

$$p_{k=\frac{\lambda}{k\mu}} p_{k-1}. \quad (5.20)$$

Используя рекуррентное соотношение (5.20) k раз, получим

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} p_0, k = 1, \dots, n \quad (5.21)$$

Для определения величины p_0 подставим (5.21) в (5.18). Тогда легко найдем.

$$p_0 = \left[\sum_{k=0}^n \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k \right]^{-1} \quad (5.21')$$

и, подставляя (5.21') в (5.21), получаем окончательно

$$p_k = \frac{\frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k}{\sum_{k=0}^n \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k}, k = 0, 1, \dots, n \quad (5.22)$$

Из формулы (5.22) следует, что вероятностные характеристики системы с потерями полностью определяются отношением интенсивности λ потока входящих требований к интенсивности μ потока уже обслуженных требований. Это отношение равно также отношению среднего времени обслуживания к среднему интервалу времени между поступлениями требований.

5.1.8. Система с остановкой. Длительность функционирования рассмотренной системы с потерями в принципе неограниченна. Предположим, что система прекращает функционирование в тот момент, когда приходит требование, а система полностью загружена. Такое состояние системы назовем поглощающим. Обозначим вероятность поглощающего состояния через $p_{n+1}(t)$. Из определения поглощающего состояния следует, что

$$p_{n+1,n}(\Delta t) = 0, p_{n+1,n+1}(\Delta t) = 1.$$

Тогда

$$p_{n+1}(t + \Delta t) = p_n(t) p_{n,n+1}(\Delta t) + p_{n+1}(t) p_{n+1,n+1}(\Delta t) + o(\Delta t) = \lambda p_n(t) + p_{n+1}(t) + o(\Delta t).$$

откуда получаем дифференциальное уравнение для определения вероятности попадания в поглощающее состояние (остановка системы)

$$p'_{n+1}(t) = \lambda p_n(t). \quad (5.23)$$

Совместно с системой дифференциальных уравнений (5.14) уравнение (5.23) определяет процесс функционирования системы до ее остановки. Так как

$$\lim_{t \rightarrow \infty} p'_{n+1}(t) = 0,$$

то из (5.23) следует, что

$$\lim_{t \rightarrow \infty} p_n(t) = p_n = 0,$$

и тогда согласно (5.21)

$$p_k = 0, 0 \leq k \leq n, \lim_{t \rightarrow \infty} p_{n+1}(t) = p_{n+1} = 1.$$

5.1.9. Система с очередями. Расширим применявшийся выше термин «состояние системы». Будем считать, что n -канальная система находится в k -м состоянии, если в ней имеется k требований: либо они все обслуживаются, если $k \leq n$, либо n требований обслуживания, а $k - n$ ждут обслуживания, если $k > n$.

Если $k < n$, то система с очередями находится в тех же условиях, что и система с потерями, так как при $k < n$ нет ни потерь, ни ожиданий. Поэтому при $k < n$ для системы с очередями остаются справедливыми все уравнения (5.14) кроме последнего.

Пусть $k \geq n$, причем длину очереди не ограничиваем. Рассуждениями, аналогичными тем, которые использовались при получении (5.7), находим

$$\begin{aligned} p_{k,k+1}(\Delta t) &= \lambda \Delta t + o(\Delta t) \\ p_{k,k+1}(\Delta t) &= 1 - \lambda \Delta t - n\mu \Delta t + o(\Delta t), \\ p_{k,k-1}(\Delta t) &= n\mu \Delta t + o(\Delta t), \\ p_{ik}(\Delta t) &= o(\Delta t), |i - k| > 1. \end{aligned}$$

По формуле полной вероятности имеем

$$p_k(t + \Delta t) = p_{k-1} \lambda \Delta t + p_k(t)(1 - \lambda \Delta t - n\mu \Delta t) + p_{k+1} n\mu \Delta t + o(\Delta t).$$

Переходя к пределу, при $\Delta t \rightarrow 0$ получаем

$$p'_k(t) = \lambda p_{k-1}(t) - (\lambda + n\mu) p_k(t) + n\mu p_{k+1}(t), k \geq n. \quad (5.24)$$

Соединяя (5.14) при $k < n$ и (5.24) при $k \geq n$, находим систему дифференциальных уравнений, определяющих вероятности состояний в системе с очередями:

$$\begin{cases} p'_0(t) = -\lambda p_0(t) + \mu p_1(t), \\ p'_k(t) = \lambda p_{k-1}(t) - (\lambda + k\mu)p_k(t) + (k+1)\mu p_{k+1}(t), 0 < k < n, \\ p'_k(t) = \lambda p_{k-1}(t) - (\lambda + n\mu)p_k(t) + n\mu p_{k+1}(t), k \geq n. \end{cases} \quad (5.25)$$

Если в начальный момент времени система свободна от требований, то начальное условие для системы (5.25) запишется в виде

$$p_0(0) = 1, p_k(0) = 0, k \geq 1. \quad (5.25')$$

Совершая над обеими частями каждого из уравнений (5.25) преобразование Лапласа, получаем с учетом начального условия (5.25') систему линейных алгебраических уравнений

$$\begin{aligned} (s + \lambda)\Phi_0(s) - \mu\Phi_1(s) &= 1, \\ -\lambda\Phi_{k-1}(s) + (s + \lambda + k\mu)\Phi_k(s) - (k+1)\mu\Phi_{k+1}(s) &= 0, 0 < k < n, \\ -\lambda\Phi_1(s) + (s + \lambda + n\mu)\Phi_k(s) - n\mu\Phi_{k+1}(s) &= 0, k \geq n, \end{aligned} \quad (5.26)$$

Где $\Phi_k(s)$ определяется согласно (5.15').

5.1.10. Предельные вероятности состояний системы с очередями. Так как при $t \rightarrow \infty$ производные $p'_k(t)$ обращаются в нуль, то имеем

$$\left. \begin{aligned} -\lambda p_0 + \mu p_1 &= 0, \\ \lambda p_{k-1} - (\lambda + k\mu)p_k + (k+1)\mu p_{k+1} &= 0, 0 < k < n, \\ \lambda p_{k-1} - (\lambda + n\mu)p_k + n\mu p_{k+1} &= 0, k \geq n \end{aligned} \right\} \quad (5.27)$$

При дополнительном условии нормирования

$$\sum_{k=0}^{\infty} p_k = 1. \quad (5.28)$$

Предельные вероятности состояний находим методом, аналогичным примененному выше. При $1 \leq k \leq n$ вводим обозначение, которое использовалось раньше [см. (5.19)]

$$y_k = \lambda p_{k-1} - k\mu p_k.$$

Тогда получим

$$y_k - y_{k+1} = 0. \quad (5.29)$$

т. е. $y_k = 0$ при $1 \leq k \leq n$ и, следовательно,

$$p_k = \frac{\lambda}{k\mu} p_{k-1} = \dots = \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} p_0.$$

или

$$p_k = \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} p_0, k \leq n.$$

Для всех $k \geq n$ вводим другое обозначение

$$y_k = \lambda p_{k-1} - n\mu p_k. \quad (5.31)$$

Тогда из последнего уравнения системы (5.27) следует $y_k = y_{k+1}$, $k \geq n$, и так как $y_n = 0$, то и $y_k = 0$ при $k \geq n$.

Следовательно, из (5.31) при $k \geq n$ находим

$$p_k = \frac{\lambda}{n\mu} p_{k-1} = \dots = \left(\frac{\lambda}{n\mu}\right)^{k-n} p_n. \quad (5.32)$$

Но из (5.30) имеем

$$p_n = \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n p_0. \quad (5.33)$$

Подставляя (5.33) в (5.32), получаем

$$p_k = \frac{1}{n!} \frac{1}{n^{k-n}} \left(\frac{\lambda}{\mu}\right)^k p_0, \quad k \geq n. \quad (5.34)$$

Используя (5.30) и (5.34), запишем условие нормирования (5.28) в виде

$$p_0 \left[\sum_{k=0}^{n-1} \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k + \frac{1}{n!} \sum_{k=n}^{\infty} \frac{1}{n^{k-n}} \left(\frac{\lambda}{\mu}\right)^k \right] = 1. \quad (5.35)$$

Если $\lambda/(n\mu) < 1$, то второе слагаемое в квадратных скобках можно преобразовать, используя формулу геометрической прогрессии:

$$\frac{1}{n!} \sum_{k=n}^{\infty} \frac{1}{n^{k-n}} \left(\frac{\lambda}{\mu}\right)^k = \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \sum_{r=0}^{\infty} \left(\frac{\lambda}{n\mu}\right)^r = \frac{1}{n!} \frac{(\lambda/\mu)^n}{1 - \lambda/(n\mu)}.$$

Тогда из (5.35) получаем выражение для вероятности p_0 :

$$p_0 = \left[\sum_{k=0}^{n-1} \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k + \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \left(1 - \frac{\lambda}{n\mu}\right)^{-1} \right]^{-1}. \quad (5.36)$$

Подставляя (5.36) в (5.30) и (5.34), находим формулы, определяющие вероятности состояний системы как при отсутствии, так и при наличии очереди. Условие применимости этих формул можно переписать в виде

$$T_v/n < T_{cp}, \quad (5.37)$$

т. е. среднее время обслуживания T_v , отнесенное к числу каналов n , должно быть меньше среднего времени T_{cp} между приходом требований. В противном случае требования от приходящего потока будут накапливаться и очередь будет неограниченно возрастать.

Вероятность ожидания для приходящего требования равна

$$p_{ож} = \sum_{k=n}^{\infty} p_k = p_0 \sum_{k=n}^{\infty} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{n! n^{k-n}} \quad (5.38)$$

где p_0 определяется согласно (5.36). Произведя в (5.38) суммирование с учетом (5.37), получим

$$p_{ож} = \frac{(1/n)(\lambda/\mu)^n}{1 - \lambda/(n\mu)} p_0 \quad (5.39)$$

или, учитывая (5.33),

$$p_{\text{ож}} = \frac{p_n}{1 - \lambda/(n\mu)}, \lambda < n\mu. \quad (5.40)$$

5.1.11. Система с ограниченной очередью. Рассмотрим систему массового обслуживания с ограничением длины очереди. Если длина становится равной m , то очередное требование теряется. Запишем систему дифференциальных уравнений, описывающих работу такой системы массового обслуживания. Пока не образовалась очередь, т. е. при $k < n+m$, для системы с ограниченной очередью остаются справедливыми все уравнения (5.25), кроме последнего. Вероятность $n+m$ -го состояния определяется из уравнения

$$p'_{n+m}(t) = \lambda p_{n+m-1}(t) - n\mu p_{n+m}(t). \quad (5.41)$$

Система дифференциальных уравнений (5.25) при $k < n+m$ и уравнение (5.41) при $k = n+m$ определяют вероятности состояний для системы массового обслуживания с ограниченной очередью.

Предельные значения вероятностей состояний (при $t \rightarrow \infty$) находятся из системы алгебраических линейных уравнений

$$\left. \begin{aligned} -\lambda p_0 + \mu p_1 &= 0, \\ \lambda p_{k-1} - (\lambda + k\mu)p_k + (k+1)\mu p_{k+1} &= 0, 0 < k < n, \\ \lambda p_{k-1} - (\lambda + n\mu)p_k + n\mu p_{k+1} &= 0, n \leq k < n+m, \\ \lambda p_{n+m-1} - n\mu p_{n+m} &= 0 \end{aligned} \right\} \quad (5.42)$$

с условием нормирования

$$\sum_{r=0}^{n+m} p_r = 1. \quad (5.43)$$

Методами, аналогичными рассмотренным выше, из этой системы алгебраических уравнений получаем

$$p_k = \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k \left[\sum_{k=0}^n \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k + \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \sum_{s=1}^m \left(\frac{\lambda}{n\mu}\right)^s \right]^{-1}, 0 \leq k \leq n, \quad (5.44)$$

$$p_k = \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \left(\frac{\lambda}{n\mu}\right)^{k-n} \left[\sum_{k=0}^n \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k + \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n \sum_{s=1}^m \left(\frac{\lambda}{n\mu}\right)^s \right]^{-1}, \quad (5.45)$$

$$n \leq k \leq n+m.$$

Вероятность потери требования равна p_{n+m} .

5.2. Восстанавливаемая система без резервирования

5.2.1. Уравнения определяющие вероятности состояний. Рассмотрим восстанавливаемую систему без резервирования, которая характеризуется постоянной интенсивностью отказов λ . После возникновения отказа система восстанавливается, причем время восстановления распределено по экспоненциальному закону с параметром μ . В периоде восстановления новые отказы не

появляются. Система может находиться в двух состояниях: работоспособном (состояние «0») и восстановления (состояние «1»). Ясно, что в рассматриваемом случае система обслуживания должна быть одноканальной, т.е. для восстановления работоспособности системы требуется одна ремонтная бригада.

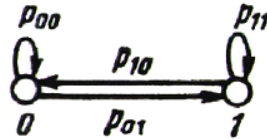


Рис. 5.2 Граф переходов состояний системы без резервирования с восстановлением

Возможны четыре вида переходов из состояния в момент времени t в состояние в момент времени $t + \Delta t$:

$0 \rightarrow 0$ (безотказная работа);

$0 \rightarrow 1$ (отказ);

$1 \rightarrow 0$ (восстановление работоспособности);

$1 \rightarrow 1$ (продолжение восстановления).

Указанные переходы можно изобразить в виде графа (рис. 5.2). Этому графу соответствует матрица переходных вероятностей размерами 2×2

$$\begin{pmatrix} p_{00}(\Delta t) & p_{01}(\Delta t) \\ p_{10}(\Delta t) & p_{11}(\Delta t) \end{pmatrix}.$$

Диагональные элементы этой матрицы равны

$$p_{00}(\Delta t) = e^{-\lambda \Delta t} = 1 - \lambda \Delta t + o(\Delta t),$$

$$p_{11}(\Delta t) = e^{-\mu \Delta t} = 1 - \mu \Delta t + o(\Delta t),$$

а элементы боковой диагонали находятся из условия равенства единице суммы элементов в каждой строке матрицы [см. (5.2)]:

$$p_{01}(\Delta t) = 1 - p_{00}(\Delta t) = \lambda \Delta t + o(\Delta t),$$

$$p_{10}(\Delta t) = 1 - p_{11}(\Delta t) = \mu \Delta t + o(\Delta t).$$

Для составления уравнений вероятностей состояний следует записать формулу полной вероятности для каждого столбца матрицы:

$$\begin{aligned} p_0(t + \Delta t) &= p_0(t)p_{00}(\Delta t) + p_1(t)p_{10}(\Delta t) = \\ &= p_0(t)(1 - \lambda \Delta t) + p_1(t)\mu \Delta t + o(\Delta t), \end{aligned}$$

$$\begin{aligned} p_1(t + \Delta t) &= p_0(t)p_{01}(\Delta t) + p_1(t)p_{11}(\Delta t) = \\ &= p_0(t)\lambda \Delta t + p_1(t)(1 - \mu \Delta t) + o(\Delta t), \end{aligned}$$

откуда получаем систему двух дифференциальных уравнений относительно вероятностей состояний «0» и «1» системы без резервирования *)

$$\begin{cases} p'_0(t) = -\lambda p_0(t) + \mu p_1(t), \\ p'_1(t) = \lambda p_0(t) - \mu p_1(t), \end{cases} \quad (5.46)$$

которая, конечно, является частным случаем системы (5.14) при $n = 1$.

5.2.2. Вероятности состояний системы, когда она работоспособна в начальный момент времени. Дополним систему уравнений (5.46) начальным условием

$$p_0(0) = 1, p_1(0) = 0, \quad (5.47)$$

т. е. будем полагать, что в начальный момент времени система находится в работоспособном состоянии. Введем изображения функций $p_0(t)$, $p'_0(t)$, $p_1(t)$, $p'_1(t)$, учитывая начальные условия (5.47)

$$\begin{aligned} p_0(t) &\rightarrow \Phi_0(s), \quad p'_0(t) \rightarrow s\Phi_0(s) - 1, \\ p_1(t) &\rightarrow \Phi_1(s), \quad p'_1(t) \rightarrow s\Phi_1(s). \end{aligned}$$

Тогда вместо системы дифференциальных уравнений (5.46) получим систему линейных алгебраических уравнений

$$(\lambda + s)\Phi_0(s) - \mu\Phi_1(s) = 1, \quad -\lambda\Phi_0(s) + (s + \mu)\Phi_1(s) = 0$$

Детерминант этой системы равен

$$\Delta = \begin{vmatrix} \lambda + s & -\mu \\ -\lambda & \mu + s \end{vmatrix} = s(s + \lambda + \mu).$$

Далее определяем

$$\Phi_1(s) = \frac{1}{\Delta} \begin{vmatrix} \lambda + s & 1 \\ -\lambda & 0 \end{vmatrix} = \frac{\lambda}{\Delta} = \frac{\lambda}{\lambda + \mu} \left(\frac{1}{s} - \frac{1}{s + \lambda + \mu} \right). \quad (5.48)$$

Переходя от изображения к оригиналу, определяем вероятность того, что в момент времени t система находится в ремонте:

$$p_1(t) = \frac{\lambda}{\lambda + \mu} [1 - e^{-(\lambda + \mu)t}]. \quad (5.48')$$

Вероятность того, что в момент времени t система работоспособна, т. е. функция готовности системы равна

$$p_0(t) = 1 - p_1(t) = \frac{\mu}{\lambda + \mu} [1 - e^{-(\lambda + \mu)t}]. \quad (5.49)$$

Из (5.49) и (5.48') при $t \rightarrow \infty$ находим предельные вероятности состояний:

*) Второе уравнение (5.46) получается непосредственно из условия, из которого следует

$$p_0 = \mu/(\lambda + \mu), \quad p_1 = \lambda/(\lambda + \mu). \quad (5.50)$$

Величина p_0 в рассматриваемом случае представляет коэффициент готовности. Ранее было доказано, что такое выражение для коэффициента готовности имеет место при любых законах распределения времени безотказной работы и времени восстановления.

Формулы (5.50) можно получить не решая системы дифференциальных уравнений, а полагая в (5.46) левые части равными нулю и $p_0(t) \rightarrow p_0$, $p_1(t) \rightarrow p_1$ при $t \rightarrow \infty$.

5.2.3. Вероятности состояний системы, когда она ремонтируется в начальный момент времени. Изменим начальные условия и предположим, что при $t = 0$ система находится в стадии восстановления, т. е.

$$p_0(0) = 0, \quad p_1(0) = 1. \quad (5.51)$$

Для начальных условий (5.51) система линейных уравнений для изображений $\Phi_0(s)$ и $\Phi_1(s)$ имеет вид

$$(\lambda + s)\Phi_0(s) - \mu\Phi_1(s) = 0, \quad -\lambda\Phi_0(s) + (\mu + s)\Phi_1(s) = 1.$$

Детерминант этой системы такой же, как и для аналогичной системы, рассмотренной в п.5.2.2:

$$\Delta = s(s + \lambda + \mu).$$

и

$$\Phi_0(s) = \frac{1}{\Delta} \begin{vmatrix} 0 & -\mu \\ 1 & s + \mu \end{vmatrix} = \frac{\mu}{\Delta} = \frac{\mu}{\lambda + \mu} \left(\frac{1}{s} - \frac{1}{s + \lambda + \mu} \right).$$

Переходя от оригинала к изображению, находим вероятность того, что в момент времени t рассматриваемая система будет работоспособна (т. е. находим функцию готовности):

$$p_0(t) = \frac{\mu}{\lambda + \mu} [1 - e^{-(\lambda + \mu)t}]. \quad (5.52)$$

На рис. 5.3 показаны графики функции готовности, когда $p_0(0) = 0$. Вероятность того, что в момент времени t система будет ремонтироваться, равна

$$p_1(t) = 1 - p_0(t) = \frac{\lambda}{\lambda + \mu} [1 - e^{-(\lambda + \mu)t}]. \quad (5.52')$$

Из (5.52) и (5.52') находим также предельные вероятности состояний:

$$p_0 = \mu/(\lambda + \mu), \quad p_1 = \lambda/(\lambda + \mu), \quad (5.53)$$

что совпадает с (5.50).

5.3. Дублирование с восстановлением

5.3.1. Основные виды задач. Рассмотрим систему, для обеспечения надежности которой используется дублирование с восстановлением. Теперь система

может находиться в трех состояниях: «0» (обе системы работоспособны), «1» (одна система восстанавливается, другая работоспособна), «2» (обе системы восстанавливаются). Соответственно возможны семь видов переходов из состояния в момент времени t в состояние в момент $t + \Delta t$:

- $0 \rightarrow 0$ (обе системы сохраняют работоспособность);
- $0 \rightarrow 1$ (отказала одна из двух систем);
- $1 \rightarrow 0$ (одна из систем восстановлена);
- $1 \rightarrow 1$ (восстановление одной системы продолжается);
- $1 \rightarrow 2$ (отказала вторая система, а первая не восстановлена);
- $2 \rightarrow 1$ (одна из двух восстанавливаемых систем восстановлена);
- $2 \rightarrow 2$ (обе системы продолжают восстанавливаться).

Указанные переходы изображены в виде графа на рис. 5.4. Этому графу соответствует матрица переходных вероятностей 3×3

$$\begin{vmatrix} p_{00}(\Delta t) & p_{01}(\Delta t) & o(\Delta t) \\ p_{10}(\Delta t) & p_{11}(\Delta t) & p_{12}(\Delta t) \\ o(\Delta t) & p_{21}(\Delta t) & p_{22}(\Delta t) \end{vmatrix}.$$

Крайние элементы побочной диагонали матрицы переходных вероятностей имеют порядок $o(\Delta t)$, так как по предположению поток отказов в системе – простейший, а время восстановления распределено по экспоненциальному закону.

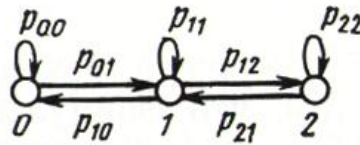


Рис. 5.4 Граф переходов состояний дублированной системы с восстановлением

Элементы первой строки матрицы переходных вероятностей зависят от режима использования резервного элемента, так как при ненагруженном режиме интенсивность потока отказов равна λ , а при нагруженном – 2λ . Поэтому

$$p_{00}(\Delta t) = 1 - (v + 1)\lambda\Delta t + o(\Delta t), \quad (5.54)$$

$$p_{01}(\Delta t) = (v + 1)\lambda\Delta t + o(\Delta t), \quad (5.54')$$

где $v = 0$ при ненагруженном режиме и $v = 1$ при нагруженном.

Элементы второй строки матрицы переходных вероятностей равны

$$p_{10}(\Delta t) = \mu\Delta t + o(\Delta t), \quad (5.55)$$

$$p_{12}(\Delta t) = \gamma \Delta t + o(\Delta t), \quad (5.55')$$

$$p_{11}(\Delta t) = 1 - p_{10}(\Delta t) - p_{12}(\Delta t) = 1 - (\gamma + \mu) \Delta t + o(\Delta t). \quad (5.55'')$$

где μ – интенсивность восстановления.

Элементы третьей строки матрицы переходных вероятностей для системы, работающей до первого отказа, равны

$$p_{21}(\Delta t) = o(\Delta t), \quad p_{22}(\Delta t) = 1 - o(\Delta t), \quad (5.56)$$

так как первое попадание в поглощающее состояние 2 означает отказ, после возникновения которого эксплуатация системы прекращается.

Элементы третьей строки матрицы переходных вероятностей для системы многократного использования зависят от числа ремонтных бригад (каналов обслуживания). При двух ремонтных бригадах отказавшее устройство сразу же начинает восстанавливаться, а при одной бригаде может возникнуть очередь на обслуживание, когда отказ одного устройства возникает во время восстановления другого. Поэтому

$$p_{22}(\Delta t) = 1 - r\mu \Delta t + o(\Delta t), \quad (5.57)$$

$$p_{21}(\Delta t) = r\mu \Delta t + o(\Delta t), \quad (5.58)$$

где r – число ремонтных бригад ($r = 1$ ил $r = 2$). (Заметим, что формулы (5.56) формально получаются из (5.57) и (5.58) при $r = 0$, но, конечно, в восстанавливаемых системах, работающих до первого отказа, имеется одна ремонтная бригада.)

Таким образом, при дублировании с восстановлением возможны 6 видов задач анализа надежности:

- системы с нагруженным резервом до первого отказа;
- системы с ненагруженным резервом до первого отказа;
- восстанавливаемые системы многократного использования с нагруженным резервом и одной ремонтной бригадой;
- восстанавливаемые системы многократного использования с нагруженным резервом и двумя ремонтными бригадами;
- восстанавливаемые системы многократного использования с ненагруженным резервом и двумя ремонтными бригадами;
- восстанавливаемые системы многократного использования с нагруженным резервом и одной ремонтной бригадой.

5.3.2. Общее решение уравнений, определяющих вероятности состояний дублированной системы с восстановлением. Используя формулу полной вероятности для каждого столбца матрицы переходной вероятности, получаем с учетом формул (5.54) – (5.58) включительно

$$\begin{aligned} p_0(t + \Delta t) &= p_0(t)[1 - (v + 1)\lambda \Delta t] + p_1(t)\mu \Delta t, \\ p_1(t + \Delta t) &= p_0(t)(v + 1)\lambda \Delta t + p_1(t)[1 - (\lambda + \mu)\Delta t] + p_2(t)r\mu \Delta t, \\ p_2(t + \Delta t) &= p_1(t)\lambda \Delta t + p_2(t)(1 - r\mu \Delta t), \end{aligned}$$

откуда находим общую для всех шести указанных выше задач систему трех дифференциальных линейных уравнений первого порядка с постоянными коэффициентами:

$$p'_0(t) = -(v+1)\lambda p_0(t) + \mu p_1(t), \quad (5.59)$$

$$p'_1(t) = (v+1)\lambda p_0(t) - (\lambda + \mu)p_1(t) + r\mu p_2(t), \quad (5.59')$$

$$p'_2(t) = \lambda p_1(t) - r\mu p_2(t), \quad (5.59'')$$

где $p_0(t)$ – вероятность того, что основная и резервная системы работоспособны в момент времени t ; $p_1(t)$ – вероятность того, что одна из систем находится в состоянии восстановления в момент времени t ; $p_2(t)$ – вероятность того, что обе системы в момент времени t неработоспособны.

Если в начальный момент времени обе системы работоспособны, то начальными условиями для системы уравнений (5.59) – (5.59'') являются

$$p_0(0) = 1, \quad p_1(0) = p_2(0) = 0. \quad (5.60)$$

Заметим также, что при любом t

$$p_0(0) = 1, \quad p_1(0) = p_2(0) = 0. \quad (5.61)$$

Для того чтобы проинтегрировать систему линейных уравнений первого порядка (5.59) – (5.59''), используем операционное исчисление. Вместо оригиналов вводим изображения с учетом начальных условий (5.60). Подвергая преобразованию Лапласа правую и левую части указанных уравнений получаем следующую систему линейных алгебраических уравнений относительно изображений:

$$\begin{aligned} s\Phi_0(s) - 1 &= -(v+1)\lambda\Phi_0(s) + \mu\Phi_1(s), \\ s\Phi_1(s) &= (v+1)\lambda\Phi_0(s) - (\lambda + \mu)\Phi_1(s) + r\mu\Phi_2(s), \\ s\Phi_2(s) &= \lambda\Phi_1(s) - r\mu\Phi_2(s) \end{aligned}$$

или

$$[s + (v+1)\lambda]\Phi_0(s) - \mu\Phi_1(s) = 1, \quad (5.62)$$

$$-(v+1)\lambda\Phi_0(s) + (s + \lambda + \mu)\Phi_1(s) - r\mu\Phi_2(s) = 0, \quad (5.62')$$

$$-\lambda\Phi_1(s) + (s + r\mu)\Phi_2(s) = 0, \quad (5.62'')$$

где

$$\Phi_k(s) = \int_0^\infty p_k(t)e^{-st}dt, \quad k = 0, 1, 2. \quad (5.63)$$

Приведем последовательность решения системы уравнений (5.62) – (5.62''). Сначала вычисляем детерминант системы

$$\Delta = \begin{vmatrix} s + (v+1)\lambda & -\mu & 0 \\ -(v+1)\lambda & s + \lambda + \mu & -r\mu \\ 0 & -\lambda & s + r\mu \end{vmatrix} =$$

$$= s\{s^2 + [(v+2)\lambda + (r+1)\mu]s + (v+1)\lambda^2 + r\mu^2 + (v+1)r\lambda\mu\} \quad (5.64)$$

и детерминант Δ_1 , который получится из Δ заменой второго столбца столбцом свободных членов системы (5.62) – (5.62''):

$$\Delta_1 = \begin{vmatrix} s + (v+1)\lambda & 1 & 0 \\ -(v+1)\lambda & 0 & -r\mu \\ 0 & 0 & s + r\mu \end{vmatrix} = (v+1)\lambda(s + r\mu). \quad (5.65)$$

Имея величины Δ и Δ_1 , можно согласно известному в высшей алгебре правилу записать выражение для функции $\Phi_1(s)$:

$$\Phi_1(s) = \frac{\Delta_1}{\Delta} = \frac{(v+1)\lambda(s+r\mu)}{\Delta}. \quad (5.66)$$

Но из (5.62'') следует

$$\Phi_2(s) = \frac{\lambda}{s + r\mu} \Phi_1(s)$$

и с учетом (5.66)

$$\Phi_2(s) = (v+1)\lambda^2 / \Delta. \quad (5.67)$$

Ясно также, что нет необходимости находить $\Phi_0(s)$, так как достаточно решить систему дифференциальных уравнений относительно $p_1(t)$ и $p_2(t)$, а $p_0(t)$ тогда вычисляется из (5.61).

Вычисляя корни уравнения $\Delta = 0$

$$s_1 = -\frac{1}{2}\{(v+2)\lambda + (r+1)\mu - \sqrt{[v\lambda - (r-1)\mu]^2 + 4\lambda\mu}\}, \quad (5.68)$$

$$s_2 = -\frac{1}{2}\{(v+2)\lambda + (r+1)\mu + \sqrt{[v\lambda - (r-1)\mu]^2 + 4\lambda\mu}\}, \quad (5.69)$$

представим изображения $\Phi_1(s)$ и $\Phi_2(s)$ в виде

$$\Phi_1(s) = \frac{(v+1)\lambda(s+r\mu)}{s(s-s_1)(s-s_2)}, \quad (5.70)$$

$$\Phi_2(s) = \frac{(v+1)\lambda^2}{s(s-s_1)(s-s_2)}. \quad (5.71)$$

Для перехода от изображений $\Phi_1(s)$, $\Phi_2(s)$ к оригиналам $p_1(t)$, $p_2(t)$ необходимо разложить дробно-рациональные функции в правых частях (5.70) и (5.71) на элементарные дроби:

$$\Phi_1(s) = \frac{(v+1)\lambda}{s_1-s_2} \left(\frac{1}{s-s_1} - \frac{1}{s-s_2} \right) + \frac{(v+1)r\lambda\mu}{s_1s_2} \times \left[\frac{1}{s} + \frac{1}{s_1-s_2} \left(\frac{s_2}{s-s_1} - \frac{s_1}{s-s_2} \right) \right], \quad (5.72)$$

$$\Phi_2(s) = \frac{(v+1)\lambda^2}{s_1s_2} \left[\frac{1}{s} + \frac{1}{s_1-s_2} \left(\frac{s_2}{s-s_1} - \frac{s_1}{s-s_2} \right) \right]. \quad (5.73)$$

Так как согласно известным соотношениям операционного исчисления

$$\frac{1}{s} \rightarrow 1, \quad \frac{1}{s-s_k} \rightarrow e^{s_k t}, \quad t \geq 0,$$

то из (5.72), (5.73) находим

$$p_1(t) = \frac{(v+1)\lambda}{s_1-s_2} (e^{s_1 t} - e^{s_2 t}) + \frac{(v+1)r\lambda\mu}{s_1s_2} \left[1 - \frac{1}{s_1-s_2} (s_1 e^{s_2 t} - s_2 e^{s_1 t}) \right], \quad (5.74)$$

$$p_2(t) = \frac{(v+1)\lambda^2}{s_1s_2} \left[1 - \frac{1}{s_1-s_2} (s_1 e^{s_2 t} - s_2 e^{s_1 t}) \right] \quad (5.75)$$

и, наконец, согласно (5.61)

$$p_0(t) = 1 - \frac{(v+1)\lambda}{s_1-s_2} (e^{s_1 t} - e^{s_2 t}) - \frac{(v+1)\lambda(r\mu+\lambda)}{s_1s_2} \left[1 - \frac{1}{s_1-s_2} (s_1 e^{s_2 t} - s_2 e^{s_1 t}) \right] \quad (5.76)$$

Заметим, что в силу известных свойств корней квадратного уравнения в формулах (5.74) – (5.76)

$$s_1s_2 = (v+1)\lambda^2 + r\mu^2 + (v+1)r\lambda\mu, \quad (5.77)$$

$$s_1 - s_2 = \sqrt{[v\lambda - (r-1)\mu]^2 + 4\lambda\mu}. \quad (5.77')$$

Предельные вероятности p_0 , p_1 , p_2 состояний системы получаются из (5.74) – (5.76) при $t \rightarrow \infty$, с учетом, что корни s_1 и s_2 – отрицательные. Эти вероятности можно получить и без предварительного решения дифференциальных уравнений, если в (5.59) – (5.59'') при $t \rightarrow \infty$ положить $p'_k(t) \rightarrow 0$, $p_k(t) \rightarrow p_k$, $k = 0, 1, 2$ и получить систему линейных алгебраических уравнений

$$\left. \begin{aligned} -(v+1)\lambda p_0 + \mu p_1 &= 0, \\ (v+1)\lambda p_0 - (\lambda + \mu)p_1 + r\mu p_2 &= 0, \\ \lambda p_1 - r\mu p_2 &= 0. \end{aligned} \right\} \quad (5.78)$$

Из (5.74) – (5.76) при $t \rightarrow \infty$ находим предельные вероятности ($s_1 \neq s_2$)

$$p_2 = \frac{(v+1)\lambda^2}{s_1s_2}, \quad p_1 = \frac{(v+1)r\lambda\mu}{s_1s_2}, \\ p_0 = 1 - \frac{(v+1)\lambda(r\mu+\lambda)}{s_1s_2} = \frac{r\mu^2}{s_1s_2}.$$

Система (5.78) имеет нулевой детерминант. Поэтому уравнения линейно зависимы. Например, второе уравнение получается как сумма первого и третьего. Рассматривая только первое и третье уравнения, получаем

$$(v+1)\lambda p_0 - \mu p_1 = 0, \quad \lambda p_1 - r\mu p_2 = 0,$$

из которых следует

$$p_1 = \frac{(v+1)\lambda}{\mu} p_0, \quad p_2 = \frac{\lambda}{r\mu} p_1 = \frac{(v+1)\lambda^2}{r\mu^2} p_0,$$

$$p_0 = \left[1 + \frac{(v+1)\lambda}{\mu} + \frac{(v+1)\lambda^2}{r\mu^2} \right]^{-1} = \frac{r\mu^2}{s_1 s_2},$$

$$p_1 = \frac{(v+1)r\lambda\mu}{s_1 s_2}, \quad p_2 = \frac{(v+1)\lambda^2}{s_1 s_2},$$

т. е. те же результаты, которые получаются предельным переходом из (5.74) – (5.76).

5.3.3. Дублированная система, работающая до первого отказа. В этом случае в общих уравнениях следует положить $r = 0$ и согласно (5.77), (5.77') получаем

$$s_1 s_2 = n\lambda^2, \quad s_1 - s_2 = \sqrt{(v\lambda + \mu)^2 + 4\lambda\mu}. \quad (5.79)$$

Используя (5.74) – (5.76) и (5.79), запишем вероятности состояний рассматриваемой системы

$$p_0(t) = \frac{1}{\sqrt{(v\lambda + \mu)^2 + 4\lambda\mu}} \{ [s_1 + (v+1)\lambda] e^{s_2 t} - [s_2 + (v+1)\lambda] e^{s_1 t} \}, \quad (5.80)$$

$$p_1(t) = \frac{(v+1)\lambda}{\sqrt{(v\lambda + \mu)^2 + 4\lambda\mu}} (e^{s_1 t} - e^{s_2 t}), \quad (5.81)$$

$$p_2(t) = 1 - \frac{1}{\sqrt{(v\lambda + \mu)^2 + 4\lambda\mu}} (s_1 e^{s_2 t} - s_2 e^{s_1 t}), \quad (5.82)$$

где

$$s_1 = \frac{1}{2} \left[(v+2)\lambda + \mu - \sqrt{(v\lambda + \mu)^2 + 4\lambda\mu} \right], \quad (5.83)$$

$$s_2 = -\frac{1}{2} \left[(v+2)\lambda + \mu + \sqrt{(v\lambda + \mu)^2 + 4\lambda\mu} \right]. \quad (5.84)$$

В приведенных формулах нагруженному резерву соответствует $v = 1$, а не-нагруженному – $v = 0$.

Вероятность безотказной работы системы равна

$$P(t) = 1 - p_2(t) = \frac{s_1 e^{s_2 t} - s_2 e^{s_1 t}}{\sqrt{(v\lambda + \mu)^2 + 4\lambda\mu}} \quad (5.85)$$

Заметим, что при $t \rightarrow \infty$ величина $p_2(t)$ стремится к единице, а величины $p_0(t)$, $p_1(t)$ — к нулю, так как корни s_1 и s_2 — отрицательные [это следует и из (5.78) при $r = 0$]. Такое поведение указанных функций закономерно, поскольку рассматривается дублированная система, работающая до первого отказа.

Среднее время безотказной работы нетрудно найти из соотношения

$$T_{cp} = \int_0^{\infty} P(t) dt = \lim_{s \rightarrow 0} P(t)e^{st} dt = \lim_{s \rightarrow 0} \left[\frac{1}{s} - \Phi_2(s) \right]. \quad (5.86)$$

Подставляя в (5.86) выражение (5.73), получаем с учетом (5.77), (5.77'), (5.83), (5.84)

$$T_{cp} = \lim_{s \rightarrow 0} \left[\frac{1}{s} - \frac{1}{s} + \frac{1}{s_1 - s_2} \left(\frac{s_2}{s - s_1} - \frac{s_1}{s - s_2} \right) \right] = \frac{s_2/s_1 - s_1/s_2}{s_1 - s_2} = -\frac{s_1 - s_2}{s_1 s_2} = \frac{(v+2)\lambda + \mu}{(v+1)\lambda^2}.$$

т. е.

$$T_{cp} = [(v+2)\lambda + \mu]/(v+1)\lambda^2. \quad (5.87)$$

Если отказавшая система не восстанавливается, т. е. $\mu = 0$, то из (5.83), (5.84) следует, что $s_1 = -\lambda$, $s_2 = -(v+1)\lambda$. Тогда для нагруженного резерва ($v = 1$) получаем

$$p_0(t) = e^{-2\lambda t}, \quad (5.88)$$

$$p_1(t) = 2(e^{-\lambda t} - e^{-2\lambda t}), \quad (5.89)$$

$$p_2(t) = 1 - 2e^{-\lambda t} + e^{-2\lambda t}, \quad (5.90)$$

$$P(t) = 2e^{-\lambda t} - e^{-2\lambda t}, \quad (5.91)$$

$$T_{cp} = \frac{3}{2\lambda}. \quad (5.92)$$

Формула (5.91) совпадает с формулой (4.62) для вероятности безотказной работы дублированной системы без восстановления при нагруженном резерве при экспоненциальном законе безотказности, а (5.92) — с формулой (4.67) для среднего времени безотказной работы этой системы.

Для ненагруженного резерва ($v = 0$) непосредственное использование приведенных выше формул для предельного случая $\mu = 0$ невозможно, так как возникает неопределенность вида $\frac{0}{0}$. Раскрывая эту неопределенность известными из математического анализа методами, получаем из (5.85) формулу (4.89) для вероятности безотказной работы дублированной системы без восстановления при ненагруженном резерве и экспоненциальном законе безотказности. Из (5.87) при $\mu = 0$ и $v = 0$ получаем

$$T_{cp} = 2/\lambda, \quad (5.93)$$

что совпадает с (4.92).

5.4 Скользящее резервирование с восстановлением

5.4.1. Постановка задачи. Предположим, что система содержит n независимых однотипных основных элементов, а также m незакрепленных резервных элементов, используемых по методу скользящего резервирования. Для восстановления отказавших элементов имеется r ремонтных бригад. Будем определять состояние системы по числу элементов, находящихся в ремонте. Задача состоит в определении вероятностей состояний системы в любой момент времени. Как и в случае дублирования, существует ряд вариантов этой задачи при следующих дополнительных предположениях:

- 1) система работает до первого отказа; при этом возможны состояния $k=0, 1, \dots, m$, а состояние $k=m+1$ является отказом (“поглощающее” состояние);
- 2) система с восстановлением используется многократно, при этом возможны состояния $k=0, 1, \dots, n+m$;

Таблица 5.2

Система	Значение λ_k для резерва	
	ненагруженного	нагруженного
До первого отказа Многократного использования при $k \leq m$	$n\lambda$	$(n+m-k)\lambda$
Многократного использования при $k > m$	$(n+m-k)\lambda$	$(n+m-k)\lambda$

нагруженный режим резервных элементов, интенсивность отказов каждого элемента постоянна и равна λ ;

ненагруженный режим резервных элементов;

число отказавших элементов меньше числа ремонтных бригад (“очереди” нет); среднее время восстановления одного элемента постоянно и равно $1/\mu$;

б) число отказавших элементов больше числа ремонтных бригад («очередь» есть).

Обозначим через μ_k параметр потока отказы системы в k -м состоянии и через μ_k – параметр потока восстановлений в этом же состоянии. Величины λ_k и μ_k в зависимости от указанных выше предположений приведены в табл. 5.2 и 5.3.

Так как поток отказов – простейший, а распределение времени восстановления – экспоненциальное, в матрице переходных вероятностей $[p_{kl}(\Delta t)]$ ненулевыми будут только элементы главной диагонали и двух соседних параллельных диагоналей ($|k-l| \leq 1$), причем

Таблица 5.3

Число отказов	μ_k
$k \leq r$	$k\mu$
$k \geq r$	$r\mu$

$$p_{k,k+1}(\Delta t) = \lambda_k \Delta t + o(\Delta t), \quad (5.117)$$

$$p_{k,k-1}(\Delta t) = \mu_k \Delta t + o(\Delta t), \quad (5.118)$$

$$\begin{aligned} p_{k,k}(\Delta t) &= 1 - p_{k,k+1}(\Delta t) - p_{k,k-1}(\Delta t) = \\ &= 1 - (\lambda_k + \mu_k) \Delta t + o(\Delta t), \end{aligned} \quad (5.119)$$

5.4.2. Система, работающая до первого отказа. По аналогии со случаем дублирования (см. п. 3.3.2 и 3.3.3) систему дифференциальных уравнений относительно вероятностей состояний рассматриваемой системы получаем из (5.117) – (5.119), используя формулу полной вероятности

$$p'_0(t) = -\lambda_0 p_0(t) + \mu_1 p_1(t), \quad (5.120)$$

$$\begin{aligned} p'_k(t) &= \lambda_{k-1} p_{k-1}(t) + (\lambda_k + \mu_k) p_k(t) + \\ &+ \mu_{k+1} p_{k+1}(t), k = 1, \dots, m-1, \end{aligned} \quad (5.121)$$

$$p'_m(t) = \lambda_{m-1} p_{m-1}(t) - (\lambda_m + \mu_m) p_m(t), \quad (5.122)$$

$$p'_{m+1}(t) = \lambda_m p_m(t), \quad (5.123)$$

где $\lambda_j, j = 0, 1, \dots, m$, определяется из табл. 5.2 в зависимости от режима использования резервных элементов, а $\mu_k = k_\mu$ при $k \leq r$ и $\mu_k = r_\mu$ при $k \geq r$.

Если в начальный момент времени все элементы системы (основные и резервные) находятся в работоспособном состоянии, то

$$p_0(0) = 1, p_k(0) = 0, k \geq 1. \quad (5.124)$$

Подвергая преобразованию Лапласа правую и левую части уравнений (5.120) – (5.123) и учитывая (5.124), получаем следующую систему алгебраических уравнений относительно изображений:

$$\begin{aligned} (s + \lambda_0) \Phi_0(s) - \mu_1 \Phi_1(s) &= 1. \\ -\lambda_{k-1} \Phi_{k-1}(s) + (s + \lambda_k + \mu_k) \Phi_k(s) - \\ -\mu_{k+1} \Phi_{k+1}(s) &= 0, k = 1, \dots, m-1, \\ -\lambda_{m-1} \Phi_{m-1}(s) + (s + \lambda_m + \mu_m) \Phi_m(s) &= 0, \\ -\lambda_m \Phi_m(s) + s \Phi_{m+1}(s) &= 0. \end{aligned} \quad (5.125)$$

Детерминант этой системы без последнего уравнения

$$\begin{aligned}
& s + \lambda_0 - \mu. \\
& \dots\dots\dots \\
D_{m+1}(s) = & -\lambda_{k-1} s + \lambda_k + \mu_k - \mu_{k-1}. \quad (5.126) \\
& \dots\dots\dots \\
& -\lambda_{m-1} s + \lambda_m + \mu_m
\end{aligned}$$

Разлагая детерминант по элементам последнего столбца, получаем рекуррентное соотношение 5

$$\begin{aligned}
D_{m+1}(s) = (s + \lambda_m + \mu_m)D_m(s) - \\
- \mu_m \lambda_{m-1} D_{m-1}(s), \quad (5.127)
\end{aligned}$$

$$D_0(s) = 1, D_1(s) = s + \lambda_0.$$

Решение системы (5.125) представляется в виде отношения

$$\Phi_i(s) = \frac{\Delta_i(s)}{D_{m+1}(s)}, i = 0, 1, \dots, m, \quad (5.128)$$

где Δ_i – детерминант, получающийся из (5.126) заменой i -го столбца столбцом, в котором первый элемент равен единице, а остальные – нулю. Изображение $\Phi_{m+1}(s)$ находится непосредственно из последнего уравнения системы (5.125)

$$\Phi_{m+1}(s) = \frac{\lambda_m}{s} \Phi_m(s). \quad (5.129)$$

Находя обратные преобразования Лапласа изображений (5.128), (5.129), получим искомые вероятности состояний системы. Вероятность безотказной работы рассматриваемой системы равна

$$P(t) = 1 - p_{m+1}(t) = \sum_{i=0}^m p_i(t). \quad (5.130)$$

5.4.3. Среднее время безотказной работы. Среднее время безотказной работы можно определить, заметив, что

$$T_{cp} = \int_0^\infty P(t)dt = \lim_{s \rightarrow 0} \int_0^\infty P(t)e^{st}dt = \lim_{s \rightarrow 0} \Phi(s). \quad (5.131)$$

Из (5.128) и (5.130), (5.131) следует

$$T_{cp} = \lim_{s \rightarrow 0} \sum_{i=0}^m \Phi_i(s) = \sum_{i=0}^m \frac{\Delta_i(0)}{D_{m+1}(0)}. \quad (5.132)$$

Можно доказать, что из (5.132) следует

⁵ Заметим, что из (5.127) следует .

$$T_{cp} = \sum_{i=0}^m \frac{1}{\lambda_i Q_i} \sum_{k=0}^i Q_k, \quad (5.133)$$

где

$$Q_k = \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \mu_2 \dots \mu_k}, Q_0 = 1. \quad (5.134)$$

Из (5.133) как частный случай при $m=n=1$ получается формула (5.87).

Для нагруженного резерва $\lambda_k = (m + n - k)\lambda$, и тогда из (5.134) находим при $\mu_k = k\mu$

$$Q_k = \left(\frac{n+m}{k} \right) \left(\frac{\lambda}{\mu} \right)^k \quad (5.135)$$

Следовательно, для нагруженного резерва среднее время безотказной работы системы равно в соответствии с (5.133)

$$\begin{aligned} T_{cp} &= \frac{1}{\lambda} \sum_{i=0}^m \sum_{k=0}^i \frac{1}{m+n-i} \left(\frac{\frac{n+m}{k}}{\frac{m+n}{i}} \right) \left(\frac{\lambda}{\mu} \right)^{k-i} = \\ &= \frac{1}{\lambda} \sum_{i=0}^m \sum_{k=0}^i \frac{i!(n+m-i-1)!}{k!(n+m-k)!} \left(\frac{\lambda}{\mu} \right)^{k-1}. \end{aligned} \quad (5.136)$$

Если нет восстановления, то $\mu \rightarrow 0$ и в двойной сумме формулы (5.136) пропадают все члены за исключением членов, соответствующих $k = i$. Тогда из (5.136) при $\mu \rightarrow 0$ имеем

$$T_{cp} = \frac{1}{\lambda} \sum_{i=0}^m \frac{1}{n+m-i} = \frac{1}{\lambda} \sum_{i=0}^m \frac{1}{k}, \quad (5.137)$$

т.е. получаем формулу для среднего времени безотказной работы для скользящего резервирования при нагруженном режиме без восстановления отказавших элементов. Формула (4.84) является частным случаем (5.137) при $n=1$. Для ненагруженного резерва $\lambda_k = n\lambda$, и тогда из (5.134) находим при $\mu_k = n\mu$

$$Q_k = \frac{1}{k!} \left(\frac{n\lambda}{\mu} \right)^k. \quad (5.138)$$

Следовательно, для ненагруженного резерва среднее время безотказной работы системы равно в соответствии с (5.133)

$$T_{cp} = \frac{1}{n\lambda} \sum_{i=0}^m \sum_{k=0}^i \frac{i!}{k!} \left(\frac{\lambda}{\mu} \right)^{k-1}. \quad (5.139)$$

Если нет восстановления, то $\mu \rightarrow 0$ и в двойной сумме остаются лишь члены, соответствующие $k = i$. Тогда

$$T_{cp} = (m + 1)/n\lambda, \quad (5.140)$$

т. е. получаем формулу для среднего времени безотказной работы для скользящего резервирования при ненагруженном режиме без восстановления. Формула (4.101) является частным случаем (5.140) при $n = 1$ (и при очевидной замене $m + 1$ на m).

5.4.4 Система с восстановлением многократного использования.

В этом случае, как отмечалось выше, возможны состояния $k=0, 1, \dots, n+m$. Поэтому система дифференциальных уравнений относительно вероятностей состояний рассматриваемой системы при начальных условиях (5.124) имеет вид

$$\left. \begin{aligned} p'_0(t) &= -\lambda_0 p_0 + \mu_1 p_1(t), \\ p'_k(t) &= \lambda_{k-1} p_{k-1}(t) - (\lambda_k + \mu_k) p_k(t) + \\ &\quad + \mu_{k+1} p_{k+1}(t), \quad k = 1, \dots, n + m - 1, \\ p'_{n+m-1}(t) &= \lambda_{n+m-1} p_{n+m-1}(t) + \mu_{n+m} p_{n+m}(t). \end{aligned} \right\} \quad (5.141)$$

Как и в предыдущем случае, эту систему можно решить при помощи преобразования Лапласа. Но при больших значениях n и m возникают трудности, связанные с вычислениями детерминантов.

В замкнутом виде легко получить выражения для предельных вероятностей состояний при $t \rightarrow \infty$. Для этого, как и в 5.1.7, положим в (5.141)

$$\lim_{t \rightarrow \infty} p_k(t) = p_k,$$

$$\lim_{t \rightarrow \infty} p'_k(t) = 0, \quad k = 0, \dots, n + m,$$

И получим систему линейных алгебраических уравнений

$$\left. \begin{aligned} -\lambda_0 p_0 + \mu_1 p_1 &= 0, \\ -\lambda_{k-1} p_{k-1} + (\lambda_k + \mu_k) p_k + \mu_{k+1} p_{k+1} &= 0, \\ k=1, \dots, n+m-1, \\ -\lambda_{n+m-1} p_{n+m-1} + \mu_{n+m} p_{n+m} &= 0, \end{aligned} \right\} \quad (5.142)$$

Обозначим

$$y_k = \lambda_{k-1} p_{k-1} - \mu_k p_k. \quad (5.143)$$

Очевидно, что $y_1 = 0$. Далее

$$y_k - y_{k-1} = \lambda_{k-1} p_{k-1} - (\lambda_k + \mu_k) p_k - \mu_{k+1} p_{k+1} = 0,$$

$$y_k = y_{k+1} = 0, \quad y_{n+m} = \lambda_{n+m-1} p_{n+m-1} -$$

$$- \mu_{n+m} p_{n+m} = 0.$$

Таким образом, для любого k имеем $y_k = 0$. Отсюда следует, что

$$p_k = \frac{\lambda_{k-1}}{\mu_k} p_{k-1},$$

$$p_k = \frac{\lambda_0 \dots \lambda_{k-1}}{\mu_1 \dots \mu_k} p_0 = Q_k p_0, \quad (5.144)$$

где использовано обозначение (5.134).

Величину p_0 определяем из условия $\sum_{k=0}^{n+m} p_k = 1$. Итак, получим

$$p_0 = (\sum_{k=0}^{n+m} Q_k)^{-1}, \quad (5.145)$$

и, следовательно,

$$p_k = Q_k / \sum_{k=0}^{n+m} Q_k, \quad k = 0, \dots, n + m, \quad (5.146)$$

Коэффициент готовности представляет собой сумму

$$K_r = \sum_{k=0}^m p_k. \quad (5.147)$$

Тогда из (5.147) с учетом (5.146) следует

$$K_r = \sum_{k=0}^m Q_k / \sum_{k=0}^{n+m} Q_k. \quad (5.148)$$

5.4.5 Частные случаи. Выпишем несколько частных случаев формул (5.146) и (5.148).

Нагруженный резерв:

а) $r=m+n$. В этом случае величины Q_k определяются согласно (5.155). Так как

$$\frac{1}{p_0} = \sum_{k=0}^{n+m} \binom{n+m}{k} \left(\frac{\lambda}{\mu}\right)^k = \left(1 + \frac{\lambda}{\mu}\right)^{n+m},$$

то

$$p_k = \binom{n+m}{k} \left(\frac{\lambda}{\mu}\right)^k \left(1 + \frac{\lambda}{\mu}\right)^{-(n+m)}, \quad k = 0, 1, \dots, m, \quad (5.149)$$

$$K_r = \left(1 + \frac{\lambda}{\mu}\right)^{-(n+m)} \sum_{k=0}^m \binom{n+m}{k} \left(\frac{\lambda}{\mu}\right)^k. \quad (5.150)$$

б) $r = 1$. В этом случае $Q_k = (\lambda/\mu)^k (n+m)! / (n+m-k)!$ и, следовательно,

$$p_k = \frac{(\lambda/\mu)^k}{(n+m-k)!} \left(\sum_{k=0}^{n+m} \frac{(\lambda/\mu)^k}{(n+m-k)!} \right)^{-1}, \quad k = 0, 1, \dots, n + m, \quad (5.151)$$

$$K_r = \sum_{k=0}^m \frac{(\lambda/\mu)^k}{(n+m-k)!} \left[\sum_{k=0}^{n-m} \frac{(\lambda/\mu)^k}{(n+m-k)!} \right]^{-1}. \quad (5.152)$$

Ненагруженный резерв:

а) $r = m + n$. В этом случае величины Q_k определяются согласно (5.138) при $k \leq m$, а при $k > m$ по формуле

$$Q_k = \frac{n^m}{k!} \left(\frac{\lambda}{\mu} \right)^k \prod_{s=1}^{k-m} (n-s),$$

Следовательно,

$$p_k = \frac{1}{k!} \left(\frac{n\lambda}{\mu} \right)^k \left[\sum_{k=0}^m \frac{1}{k!} \left(\frac{n\lambda}{\mu} \right)^k + n^n \sum_{k=m+1}^{m+n} \frac{1}{k!} \left(\frac{\lambda}{\mu} \right)^k \prod_{s=1}^{k-m} (n-s) \right]^{-1}, \quad (5.153)$$

$$k \leq m;$$

$$P_k = n^m \left(\frac{\lambda}{\mu} \right)^k \prod_{s=1}^{k-m} (n-s)^*$$

$$* \left[\sum_{k=0}^m \frac{1}{k!} \left(\frac{n\lambda}{\mu} \right)^k + n^m \sum_{k=m+1}^{m+n} \frac{1}{k!} \left(\frac{\lambda}{\mu} \right)^k \prod_{s=1}^{k-m} (n-s) \right]^{-1}, \quad (5.154)$$

$$k > m;$$

$$K_r = \sum_{k=0}^m \frac{1}{k!} \left(\frac{n\lambda}{\mu} \right)^k \left[\sum_{k=0}^m \frac{1}{k!} \left(\frac{n\lambda}{\mu} \right)^k + n^m \sum_{k=m+1}^{m+n} \frac{1}{k!} \left(\frac{\lambda}{\mu} \right)^k \prod_{s=1}^{k-m} (n-s) \right]^{-1} \quad (5.155)$$

б) $r = 1$.

$$p_k = \left(\frac{n\lambda}{\mu} \right)^k \left[\sum_{k=0}^m \left(\frac{n\lambda}{\mu} \right)^k + n^m \sum_{k=m+1}^{m+n} \left(\frac{\lambda}{\mu} \right)^k \prod_{s=1}^{k-m} (n-s) \right]^{-1}, \quad (5.156)$$

$$k \leq m;$$

$$p_k = n^m \prod_{s=1}^{k-m} (n-s) \left(\frac{\lambda}{\mu} \right)^k \left[\sum_{k=0}^m \left(\frac{n\lambda}{\mu} \right)^k + n^m \sum_{k=m+1}^{m+n} \left(\frac{\lambda}{\mu} \right)^k \prod_{s=1}^{k-m} (n-s) \right]^{-1} \quad (5.157)$$

$$k > m;$$

$$K_r = \sum_{k=0}^m \left(\frac{n\lambda}{\mu}\right)^k \left[\sum_{k=0}^m \left(\frac{n\lambda}{\mu}\right)^k + n^m \sum_{k=m+1}^{m+n} \left(\frac{\lambda}{\mu}\right)^k \prod_{s=1}^{k-m} (n-s) \right]^{-1}. \quad (5.158)$$

5.4.6. Пример. Воспользуемся формулами п. 5.4.5 для определения коэффициента готовности радиорелейной линии, в которой $n=6$ основных и $m=2$ резервных стволов. Предположим, что для каждого ствола $\lambda/\mu=0,1$. Тогда при нагруженном резервировании из (5.150) и (5.152) и находим:

при шести ремонтных бригадах

$$K_r = \frac{1}{1,1^6} \left(1 + \left(\frac{6}{1}\right) 0,1 + \left(\frac{6}{2}\right) 0,01 \right) \approx 0,988,$$

при одной ремонтной бригаде

$$K_r = \frac{\frac{1}{6!} + \frac{1}{5!} 0,1 + \frac{1}{4!} 0,01}{\frac{1}{6!} + \frac{1}{5!} 0,1 + \frac{1}{4!} 0,01 + \frac{1}{3!} 0,001 + \frac{1}{2!} 10^{-4} + 10^{-5} + 10^{-6}} \approx 0,873.$$

При ненагруженном резервировании из (5.155) получим:

при шести ремонтных бригадах

$$K_r = \frac{1 + 0,6 + 0,5 + 0,36}{1 + 0,6 + 0,5 * 0,36 + 36 \left(\frac{5}{3!} 10^{-3} + \frac{5 * 4}{4!} 10^{-4} + \frac{5 * 4 * 3}{5!} 10^{-5} + \frac{5 * 4 * 3 * 2}{6!} 10^{-6} \right)} \approx 0,998$$

при одной ремонтной бригаде

$$K_r = \frac{1 + 0,6 + 0,5 + 0,36}{1 + 0,6 + 0,36 + 36(5 * 10^{-3} + 5 * 4 * 10^{-4} + 5 * 4 * 3 * 10^{-5} + 5 * 4 * 3 * 2 * 10^{-6})} \approx 0,888.$$

6. КУБИТНЫЕ (КВАНТОВЫЕ) ТЕХНОЛОГИИ ПОВЫШЕНИЯ НАДЕЖНОСТИ МЭМС

6.1. Кубитные структуры данных вычислительных устройств восстановления работоспособности

Рыночная привлекательность эмуляции квантовых методов вычислений при создании виртуальных (облачных) компьютеров (ВК) в киберпространстве основывается на использовании кубитных моделей данных, ориентированных на параллельное решение задач дискретной оптимизации за счет существенного повышения затрат памяти. Здесь не рассматриваются физические основы квантовой механики, касающиеся недетерминированного взаимодействия атомных частиц [23-26], но используется понятие кубита как двоичного или многозначного вектора для совместного и одновременного задания булеана состояний дискретной области киберпространства на основе линейной суперпозиции унитарных кодов, ориентированных на параллельное исполнение методов анализа и синтеза компонентов киберпространства.

Мотивация нового подхода для проектирования ВК связана с появлением облачных сервисов, которые представляют собой специализированные и рассредоточенные в пространстве виртуальные компьютерные системы, инвариантные по отношению к реализации в аппаратуре или в программном продукте. Иначе, сегодня уже не всегда интересно программисту как и куда имплементируются коды программных приложений, равно как и нет необходимости знать теорию проектирования цифровых автоматов на уровне вентилей, регистровых передач и использовать специфические компоненты вычислительной техники (триггеры, регистры, счетчики, мультиплексоры). Любой компонент функциональности покрывается векторной формой таблицы истинности, реализуемой с помощью памяти. Логические функции в традиционном исполнении здесь исключены из рассмотрения. От этого частично страдает быстродействие, но учитывая, что 94 процента SoC-кристалла сегодня есть память, то остальные 6 процентов также можно реализовать на памяти, что не будет критичным для большинства программных и аппаратных приложений. Поэтому практически полезной для программирования эффективных виртуальных компьютеров будет теория, основанная на двух, более высокого уровня абстракции, компонентах: память и транзакция.

Особенность организации данных в классическом компьютере заключается в том, что каждый бит, байт или другой компонент имеет свой адрес. Поэтому существует проблема в эффективной обработке ассоциации (конечного алфавита символов) равных по значимости элементов, которые не имеют порядка по определению, например, множество всех подмножеств. Решением может быть процессор, где элементарной ячейкой служит образ или шаблон

универсума из n унитарно кодированных примитивов, которые используют

суперпозицию для формирования булеана $Q = 2^n$ всех возможных состояний такой ячейки в виде множества всех подмножеств [27, 28]. Прослеживается определенная аналогия по структурам данных с квантовым компьютером, где понятию кубита в классическом компьютере можно поставить во взаимно однозначное соответствие четыре состояния: $\{10, 01, 11, 00\}$ [29] век-

торного задания булеана примитивов Кантора $A^k = \{0, 1, X, \bar{X}\}$, где X – описывает «перепутанное», одновременно суперпозиционирующее состояние двух уровней сигналов 0 и 1. Большинство работ, в частности [23-26], рассматривает возможность эмуляции классических вычислительных процессов на квантовых компьютерах, но учитывая «реверсность» или обратимость упомянутого соответствия, далее предлагается обратное преобразование – эмуляция некоторых преимуществ квантовых вычислений на классических процессорах.

На рынке электронных технологий существует конкуренция между базами имплементации идеи [30]: 1) Гибкая (мягкая) реализация проекта связана с синтезом интерпретативной модели программной реализации устройства или в аппаратном исполнении программируемых логических устройств на основе FPGA, CPLD. Здесь преимущества заключаются в технологичности модификации проекта, недостатки – в невысоком быстродействии функционирования цифровой системы; 2) Жесткая реализация имеет ориентацию на использование компилятивных моделей при разработке программных приложений или на имплементацию проекта в кристаллах VLSI. Преимущества и недостатки жесткой реализации инверсны по отношению к мягкому исполнению проектов: высокое быстродействие и невозможность модификации. С учетом четырех изложенных базовых вариантов для реализации идеи ниже предлагаются квантовые структуры данных, ориентированные на повышение быстродействия гибких моделей программного или аппаратного исполнения проекта.

6.1.1. Квантовые структуры описания цифровых систем

n -Кубит есть векторная форма унитарного кодирования универсума из n

примитивов для задания булеана состояний 2^{2^n} с помощью 2^n двоичных переменных. Например, если $n=2$, то 2-кубит задает 16 состояний с помощью 4-х переменных. Если $n=1$, то кубит задает 4 состояния на универсуме из двух примитивов с помощью 2-х двоичных переменных (00, 01, 10, 11) [23, 30]. При этом допускается суперпозиция (одновременное существование) в

векторе 2^n состояний. Кубит (n -кубит) дает возможность использовать логические операции вместо теоретико-множественных для существенного ускорения процессов анализа дискретных систем. Далее кубит отождествляется с n -кубитом или вектором, если это не мешает пониманию излагаемого материала. Поскольку квантовые вычисления связаны с анализом кубитных

структур данных, то далее частично эксплуатируется определение «квантовый» для идентификации технологий, использующих два свойства квантовой механики: параллелизм обработки и суперпозицию состояний. Синонимом кубита при задании логической функциональности является: Q-покрытие (Q-вектор) [29], как унифицированная векторная форма суперпозиционного задания выходных состояний, соответствующих унитарным кодам адресов входных переменных любой логической функции.

Кубит в цифровой системе выступает в качестве формы задания структурного примитива, инвариантной к технологиям реализации функциональности (hardware, software). Более того, «квантовый» синтез цифровых систем на основе кубитных структур уже не привязан жестко к теореме Поста, определяющей условия существования функционально полного базиса. На предлагаемом уровне абстракции n-кубит дает исчерпывающие и более широкие возможности для векторного задания любой функции из множества

$$b(f) = 2^n.$$

Формат структурного кубитного компонента цифровой схемы $Q^* = (X, Q, Y)$ включает интерфейс (входные и выходную переменные), а также кубит-вектор Q , задающий функциональность $Y = Q(X)$, размерность которого определяется степенной функцией от числа входных линий $k = 2^n$.

Практически ориентированная новизна кубитного моделирования заключается в замене таблиц истинности компонентов цифрового устройства векторами состояний выходов. Достаточно просто можно продемонстрировать такие преобразования, применительно к логическому элементу. Пусть функциональный примитив имеет следующее двоичное покрытие:

$$P = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ \hline \end{array}$$

которое можно трансформировать путем унитарного кодирования входных векторов на основе использования двухтактного алфавита [30, 31]. Изначально он был предназначен для компактного описания всех возможных переходов автоматных переменных, что иллюстрируется на рис. 6.1 соответствующим графом и интерпретацией символов.

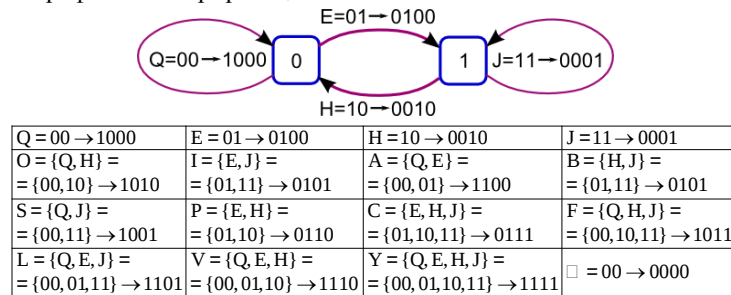


Рис. 6.1. Двухтактный алфавит автоматных переменных

Здесь представлены: символы, их двоичные и унитарные коды (например, Q = 00 – 1000), предназначенные для описания двух соседних состояний автоматных переменных. Структурно алфавит представляет собой булеан (множество всех подмножеств) состояний на универсуме из четырех примитивов $Y = \{Q, E, H, J\}$. Унитарный код соответствует формату вектора, содержащего два кубита, с помощью которых формируются 16 символов двухтактного алфавита. Используя последний любое покрытие функционального двухвходового логического примитива можно представить двумя кубами или даже одним, учитывая, что они взаимно инверсны:

$$P = \begin{bmatrix} 00 & 1 \\ 01 & 1 \\ 10 & 1 \\ 11 & 0 \end{bmatrix} = \begin{bmatrix} Q & 1 \\ E & 1 \\ H & 1 \\ J & 0 \end{bmatrix} = \begin{bmatrix} V & 1 \\ J & 0 \end{bmatrix} = \begin{bmatrix} 1110 & 1 \\ 0001 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$$

Здесь сначала кодируются все пары символами двухтактного алфавита, затем выполняется объединение первых трех кубов по правилу оператора со-граней [27]: векторы отличающиеся в одной координате минимизируются в один. Далее полученное покрытие из двух кубов кодируется соответствующими данным символам кубитными векторами. Для моделирования исправного поведения достаточно иметь только один куб (нулевой или единичный), поскольку второй всегда является дополнением к первому. Следовательно, ориентируясь, например, на единичный куб, формирующий на выходе 1, можно убрать бит состояния выхода примитива, что уменьшит размерность куба или модели примитива до количества адресуемых состояний элемента, где адрес есть вектор, составленный из двоичных значений входных переменных, по которому определяется состояние выхода примитива. Ввиду тривиальности нет смысла показывать, что по аналогии любую таблицу истинности можно привести к кубитной функциональности в форме вектора выходных состояний логического элемента, имеющего n входов.

Процедура моделирования на Q-векторе функциональности сводится к записи в выходную переменную Y состояния бита, адрес которого сформирован на основе конкатенации значений входных переменных:

$$Y = Q(X) = Q(X_1 * X_2 \dots * X_j \dots * X_k)$$

Для моделирования цифровых систем, где компонентами выступают взаимосвязанные на основе M-вектора эквипотенциальных линий Q-примитивы, процедура обработки последних определяется выражением:

$$M(Y) = Q[M(X)] = Q[M(X_1 * X_2 \dots * X_j \dots * X_k)]$$

С учетом сквозной нумерации Q-примитивов универсальная процедура моделирования текущего i -элемента будет иметь формат:

$$M(Y_i) = Q_i[M(X_i)] = Q_i[M(X_{i1} * X_{i2} \dots * X_{ij} \dots * X_{ik_i})]$$

В данном случае существен-

но упрощается алгоритм анализа цифровой системы и в 2^n раз повышается быстродействие интерпретативного моделирования за счет увеличения объема памяти для описания функциональности схемной структуры.

Синтез Q-покрытия цифровой системы сводится к выполнению операции суперпозиции над Q-векторами функциональностей, входящих в нее. На-

пример, для трех примитивов (элементы and, and-not, and-not), составляющих схему, операция суперпозиции формирует Q-вектор всей функциональности, где его размерность будет больше, чем сумма Q-покрытий исходных примитивов:

$$\begin{array}{|c|c|c|c|c|c|} \hline a & 0 & 0 & 0 & 1 & c \\ \hline b & & & & & \\ \hline d & 1 & 1 & 1 & 0 & f \\ \hline e & & & & & \\ \hline \end{array} \begin{array}{|c|c|c|c|c|c|} \hline c & 1 & 1 & 1 & 0 & g \\ \hline f & & & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & g \\ \hline b & & & & & & & & & & & & & & & \\ \hline d & & & & & & & & & & & & & & & \\ \hline e & & & & & & & & & & & & & & & \\ \hline \end{array} .$$

Но при этом процедура моделирования Q-вектора структуры будет иметь более высокое быстродействие, поскольку она представлена только одним обращением к Q-покрытию для выемки содержимого из ячейки вместо трех, когда система представлена тремя примитивами.

Трехэлементная схема, представленная выше Q-векторами, может быть задана схемотехнически (удобно для человека), где вместо векторов будут фигурировать соответствующие десятичные номера:

$$\begin{array}{|c|c|c|} \hline a & 1 & c \\ \hline b & & \\ \hline d & 14 & f \\ \hline e & & \\ \hline \end{array} \begin{array}{|c|c|c|} \hline c & 14 & g \\ \hline f & & \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & 34679 & g \\ \hline b & & \\ \hline d & & \\ \hline e & & \\ \hline \end{array} .$$

При обработке такой формы функциональных покрытий необходимо «развернуть» десятичный код в двоичный вектор и вычислить адрес ячейки, содержимое которой будет определять состояние выходной переменной, в данном случае g. Естественно, что десятичный код существует на бумаге, а в компьютере это представление – всегда двоичный вектор. На самом деле «мягкая» схемотехника идентификации (нумерации) межсоединений имеет будущее, поскольку не связана с соединительными проводами, которые заменяются адресами или номерами линий, создающими структуру цифрового изделия, которая обладает гибкостью замены примитивов в случае обнаружения ошибок проектирования или дефектов.

Кубитное представление функциональных элементов дает также возможность ввести новые схемотехнические обозначения, связанные с десятичным номером Q-вектора, задающего функциональность. Если система логических элементов имеет $n=2$ входа, то число всех возможных функций равно

$k = 2^{2^n}$, где типы или номера функционалов представлены в нижней строке следующей таблицы:

00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
f =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Более того, на основе множества кубитов первого уровня, задающих функции от двух переменных, можно ввести кубит второго уровня, унитарно ко-

дирующий двухходовые функции, что дает возможность создавать структуру одновременного задания и анализа всех неупорядоченных состояний дискретной системы, где входными переменными выступают уже функционалы первого уровня:

00	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Q=	0	1	1	1	1	1	0	0	0	0	0	0	1	1	0

В данной таблице представлены четыре вектора-примитива входных переменных (00,01,10,11), образующие $k = 2^{2^2} = 2^4$ полное множество всех возможных функций, которые рассматриваются в качестве примитивов второго уровня. Затем – векторы-примитивы выходных переменных (16 столбцов от

0000 до 1111), формируют уже $k = 2^{2^4} = 2^{16}$ функциональных примитивов, входящих в состав более сложной дискретной системы, которые можно анализировать параллельно! Далее можно экстраполировать создание более сложной системы кубитов, где вектор $Q=0111110000001100$, представлен-

ный нижней строкой, будет рассматриваться как один из $k = 2^{2^{16}}$ примитивов третьего уровня иерархии. В каждом уровне иерархии кубитов количество или булеан состояний экспоненциально зависит от числа примитивов-

векторов $k = 2^{2^n}$. Если вектор Q имеет все единичные значения $Q=1111111111111111$, то он одновременно определяет пространство, содержащее 16 символов двухтактного алфавита, которые соответствуют булеану на универсуме из четырех примитивов [31].

Основная инновационная идея квантовых вычислений по сравнению с машиной фон Неймана заключается в переходе от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение (точку) к квантовым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений. В этом тезисе сформулировано будущее всех высокопроизводительных компьютеров для параллельного нецифрового анализа структур и сервисов дискретного киберпространства. Иначе, вычислительная сложность выполнения процедуры обработки множества из n элементов в «квантовом» процессоре и одного в машине фон Неймана равны между собой за счет соответствующего n -кратного повышения аппаратной сложности «квантовой» структуры.

6.1.2. Графовые структуры описания цифровых схем

Несколько отличная схемотехника, не привязанная непосредственно к транзисторам, может быть представлена графовыми структурами, где каждая вершина (дуга) отождествляется с функциональным преобразованием, которое задается Q -вектором. Тогда дуга (вершина) определяет

взаимосвязи между функциональными Q-покрытиями, а также входные и выходные переменные. Вопрос реализации таких структур изначально привязан к ячейкам памяти (LUT FPGA), которые способны хранить информацию в виде Q-вектора, где каждый бит или разряд имеет свой адрес, отождествляемый со входным словом. Тем не менее, программная реализация таких структур становится конкурентоспособной по быстродействию на рынке промышленных систем проектирования цифровых систем на кристаллах за счет адресной реализации процессов моделирования функциональных примитивов. Кроме того, аппаратная поддержка систем проектирования в виде Hardware Embedded Simulator (HES, Aldec) [32] приобретает новую мотивацию на системном уровне проектирования цифровых изделий, когда программные и аппаратные решения имеют один и тот же кубитный формат. Далее для рассмотрения предлагается комбинационная схема (рис. 6.2), содержащая 6 примитивов и три различных логических элемента. Данной схеме соответствуют три универсальные графовые формы цифровой функциональности (рис. 6.3), использующие Q-векторы для задания поведения логических примитивов.

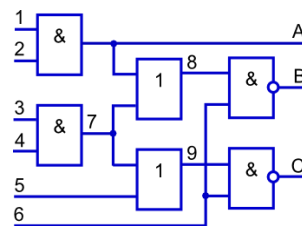


Рис. 6.2. Комбинационная структура логических примитивов

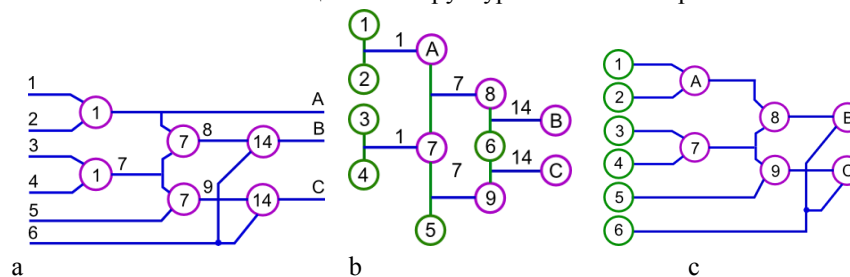


Рис. 6.3. Графовые формы квантовых функциональностей

Структура, представленная на рис. 6.3а содержит 12 линий (дуг), нагруженных на квантовые функциональности (1 = 0001, 7=0111, 14=1110). Она подобна традиционной структурно-функциональной модели комбинационной схемы. Граф на рис. 6.3b напоминает модель регистровых передач С.Г. Шаршунова [33], который является обратным по отношению к первой структуре. Здесь горизонтальные синие дуги отождествляются с функциональностями, а вершины – с группами входных для функциональностей линий, объединенных в регистровые переменные посредством зеленых вертикальных дуг, состояния которых образуют двоичный вектор, используемый в качестве адреса для вычисления состояния логического элемента или более

сложного функционала. Переменные, участвующие в формировании адреса для Q-вектора функциональности можно объединить в одну вершину с указанием всех идентификаторов линий, создающих вектор-адрес. Регистровый граф комбинационной схемы является ранжированным по уровням формирования входных сигналов, что обеспечивает условия параллелизма обработки элементов одного уровня и выполнимости итераций Зейделя [27–31], повышающих быстродействие алгоритмов исправного моделирования цифровых систем. Структура на рис. 6.3b интересна своей регистровой реализацией, что можно использовать для формализации как программных, так и аппаратных моделей вентильного, регистрового и системного уровней. Такое представление трудно воспринимается человеком, но оно технологично и легко понимается компьютером для автоматического создания программных систем анализа вычислительных структур и сервисов киберпространства. Таким образом, регистровый граф цифровой схемы представляют собой разрывную по гальваническим связям, а потому гибкую систему взаимосвязанных адресных примитивов для формирования функциональной структуры любой сложности, прежде всего в масштабах PLD, где все комбинационные примитивы представлены постоянными запоминающими устройствами (LUT), что обеспечивает высокое быстродействие функционирования и ремонт логических модулей в режиме online.

Одномерный Q-вектор описания функциональности можно привязать к выходной (внутренней) линии устройства, состояние которой формируется в процессе моделирования рассматриваемого Q-покрытия. Тогда регистровая реализация комбинационного устройства может быть представлена вектором моделирования M, на невходные линии которого нагружена функциональность с дугами, идущими от входных переменных, значения которых задают адрес бита Q-вектора, формирующего состояние рассматриваемой невходной линии (рис. 6.3c). Иначе, если функциональности описываются одновыходовыми примитивами, то каждый из них можно отождествить или идентифицировать номером или координатой невходной линии, на которую нагружен данный элемент. Если функциональность многовыходовая, то Q-покрытие уже представляется матрицей с количеством строк, равным числу выходов. Эффект от такого примитива заключается в параллелизме одновременного вычисления состояний нескольких выходов за одно обращение к матрице по текущему адресу! Данное обстоятельство является существенным аргументом в пользу синтеза обобщенных кубитов для фрагментов цифрового устройства или всей схемы в целях их параллельной обработки в одном временном такте. Близкой к идеальной по компактности и времени обработки структурой данных, где Q-векторы функциональностей и номера входных переменных привязаны к невходным линиям устройства, является следующая таблица:

L	1	2	3	4	5	6	7	8	9	A	B	C
M	1	1	1	1	1	0	0	1	1	1	0	1
X	.	.	.	.	.	.	34	A7	75	12	86	96
Q	.	.	.	.	.	.	0	0	0	0	1	1
	.	.	.	.	.	.	0	0	1	1	1	1
	.	.	.	.	.	.	0	0	1	1	1	1
	.	.	.	.	.	.	1	1	1	1	0	0

Она дает представление о том, какие переменные цифровой схемы являются внешними, сколько функциональных примитивов имеется в структуре, а также какие входы нагружены на каждый Q-вектор. Достоинством таблицы является отсутствие вектора номеров выходов для каждого примитива, но при этом сохраняется необходимость иметь номера входных переменных для формирования адресов, манипулирование которыми есть достаточно затратный процесс. Модель анализа схемной структуры упрощается до вычисления двух адресов (!) при формировании вектора моделирования:

$M_i = Q_i[M(X_i)]$ путем исключения сложного адреса выхода примитива в процессе записи состояний выходов в координаты M-вектора.

Кубитно-регистровый граф из рис. 6.3с может быть представлен в виде матрицы

$m = \{m_{ij}\}$, $i = \overline{1, p}$; $j = \overline{1, q}$ параллельно-последовательной обработки логических примитивов:

μ_{ij}	1	2	3
1	$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ A \end{bmatrix}$	$\begin{bmatrix} A \\ 7 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$	$\begin{bmatrix} 8 \\ 6 \end{bmatrix} \begin{bmatrix} 14 \\ B \end{bmatrix}$
2	$\begin{bmatrix} 3 \\ 4 \end{bmatrix} \begin{bmatrix} 1 \\ 7 \end{bmatrix}$	$\begin{bmatrix} 7 \\ 5 \end{bmatrix} \begin{bmatrix} 7 \\ 9 \end{bmatrix}$	$\begin{bmatrix} 6 \\ 9 \end{bmatrix} \begin{bmatrix} 14 \\ C \end{bmatrix}$
3	$\begin{bmatrix} X \\ X \end{bmatrix} \begin{bmatrix} 1 \\ X \end{bmatrix}$	$\begin{bmatrix} X \\ X \end{bmatrix} \begin{bmatrix} 7 \\ X \end{bmatrix}$	$\begin{bmatrix} X \\ X \end{bmatrix} \begin{bmatrix} 14 \\ X \end{bmatrix}$

который отображает взаимодействие Q-покрытий по трем уровням срабатывания относительно формата (X-Q-Y) входы- Q-вектор -выход каждого примитива: [(1,2-1-A), (3,4-1-7)], [(A,7-7-8), (7,5-7-9)], [(8,6-14-B), (6,9-14-C)]. Чтобы корректно отработала функциональность примитива, необходимо к рассматриваемому моменту сформировать все его входные переменные. Поэтому кубитно-регистровый граф разделен на уровни срабатывания, где все примитивы внутри одного уровня могут обрабатываться параллельно, а сами уровни – последовательно друг за другом. Кубитная матрица своей регулярной структурой ориентирована на решение задач: 1) Ремонт логических примитивов в процессе функционирования за счет переадресации дефектных элементов на примитивы из запаса (строка 3) [34], подобно тому, как это делается в матричной памяти; 2) Индексная адресация каждого кванта

матрицы $\mu_{ij} \in \mu$, $\mu_{ij} = (X_{ij}, Q_{ij}, Y_{ij})$ для оперативного ремонта отказавших примитивов (в примере можно заменить три дефектных примитива, по одному из каждого слоя); 3) Обеспечение высокого быстродействия прототипа комбинационного устройства, реализованного на основе кубитных примитивов, которые имплементируются на кристалле PLD в LUT-элементы

[8], за счет предоставления возможности параллельной обработки элементов одного слоя; 4) Создание матричного кубитного мультипроцессора, ориентированного на анализ аппаратных прототипов комбинационных устройств большой размерности, позволяющих существенно ускорить процессы тестирования и верификации цифровых систем на кристаллах, как это делает Hardware Embedded Simulator (HES), компании Aldec [10]; 5) Разработка методов анализа комбинационных схем, ориентированных на матричное исполнение кубитных структур логических элементов путем их имплементации в элементы памяти кристаллов PLD; 6) Создание генератора кода для масштабируемого синтеза квантовых матриц комбинационных схем на основе использования структур схемотехнических примитивов кристаллов PLD; 7) Проектирование управляющего автомата для функциональной обработки и сервисного обслуживания (восстановления работоспособности) кубитной матрицы комбинационного устройства, имплементированного в PLD структуру.

Модель управляющего автомата для симулирования кубитной структуры комбинационной схемы укладывается в три пункта:

1. Инициирование очередного входного воздействий для комбинационного устройства.
2. Выбор очередного слоя (столбца матрицы) с номером i для параллельной обработки кубитных примитивов Q в целях формирования состояний выходов по адресу входного слова, представленного вектором $M(X_{ij})$, где X_{ij} – вектор номеров входных переменных для примитива Q_{ij} , M – вектор моделирования всех линий комбинационного устройства:

$$M(Y_{ij}) = Q_{ij}[M(X_{ij})], j = \overline{1, q}.$$
3. Приращение индекса столбца $i=i+1$ и переход к пункту 2 обработки очередного слоя кубитных примитивов. По окончании анализа всех столбцов матрицы $i = p$ выполняется инкремент индекса очередного входного воздействия $t=t+1$ с последующим переходом на пункт 1. При достижении конечного числа входных наборов $t = n_{\max}$ цикл обработки теста для кубитной матрицы заканчивается.

6.1.3. Автоматная структура MQT-процессора

Основой процесс-модели функционирования вычислителя (комбинационной схемы), представленного в форме кубитной матрицы, является операция транзакции (считывания-записи) двоичной информации на LUT-элементах памяти структуры PLD: $M(Y) = Q[M(X)]$, которая формирует сколь угодно сложные функциональности и сервисы. Иначе, все вычислительные процессы в компьютерных системах, сетях и в киберпространстве можно свести к одной операции транзакции на любой структуре, способной хранить ин-

формацию. Все технологические и схмотехнические узлы, на которых реализуются аппаратные и программные продукты, можно не принимать во внимание, чтобы синтезировать виртуальные информационные сервисы, где следует использовать только операцию запись-считывание, как базовую процедуру над Q-покрытиями, которые своей гибкой универсальностью поглощают все конструктивы вычислительной техники для синтеза и анализа объектов, процессов и явлений в киберпространстве. Появляется новая MQT-модель «мягкого» кубитного виртуального компьютера, основанная на транзакции (Transaction) между компонентами памяти Q (Qubit), объединенными в систему с помощью вектора моделирования M (Memory). Естественно, что любые логические функции реализуются путем считывания битов Q-вектора $Q = (q_1, q_2, \dots, q_i, \dots, q_n), q_i \in \{0, 1\}$. Но при этом становится несколько избыточной теорема Поста, формирующая условия в виде математического функционального базиса логических функций, необходимо и достаточно полного для создания любой вычислительной системы, поскольку Q-вектор есть универсальная, гибкая и компактная форма описания как простой, так и сложной функциональности киберпространства. Транзакция может существовать только при наличии заданных отношений на компонентах памяти, число которых должно быть не менее одного. Цикличность транзакции $M \xleftarrow{Y_i} Q_i \xleftarrow{X_i} M$, (рис. 6.4a, 6.4b) представленной Read-Write операциями $M(Y_i) = Q_i[M(X_i)]$ в MQT-структуре, определяется использованием вектора M взаимных связей компонентов, который образует «мягкую», адресно ориентированную и гальванически разорванную автоматную модель, где главным компонентом вычислительной системы фигурирует кубит-память:

$$\begin{cases} A = \langle M, Q, f, g, X, Y \rangle, \\ M(t+1) = f[X(t), Q(t), M(t)]; \\ Y(t) = g[X(t), Q(t), M(t)]; \\ M(Y_i) = Q_i[M(X_i)]. \end{cases}$$

Данные выражения подходят под классическое определение автомата Мура, но здесь равенство $M(Y_i) = Q_i[M(X_i)]$ задает основную и единственную процедуру анализа адресуемых Q-компонентов цифровой системы для формирования координат вектора состояния M. Последний представляет собой адресно вычисляемые реакции Q-примитивов устройства на входные воздействия, полученные конкатенацией координат вектора M, адреса которых соответствуют номерам входных переменных рассматриваемого примитива. Автомат имеет входную регистровую переменную X, вектор выходных сигналов Y, который являясь подмножеством переменных вектора M, может быть исключен из рассмотрения. Адресные регистровые переменные Y_i, X_i каждого примитива дают возможность формировать состояние вектора M путем моделирования структуры квантовых примитивов Q.

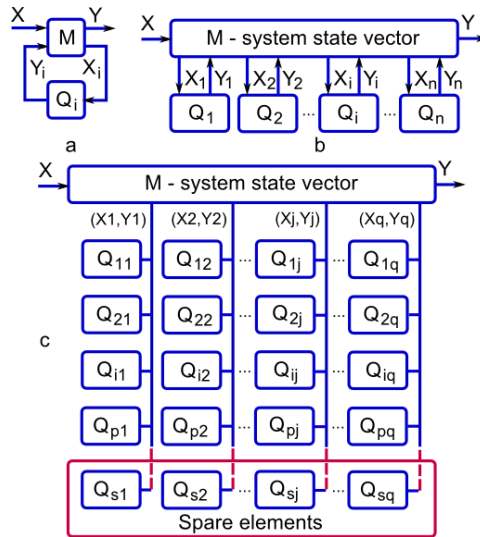


Рис. 6.4. Битовая, векторная и матричная структуры кубитных примитивов

Здесь (рис. 6.4) модель цифрового изделия в виде структурного MQT-автомата включает вектор моделирования или состояния системы, который объединяет и «мягко» структурирует все кубитные примитивы для формирования функциональности, заданной спецификацией. При этом цикл обработки каждого Q-элемента заключается в транзакционном взаимодействии между собой пары компонентов M–Q (рис. 6.4a), которые в совокупности реализуют универсальную функциональность $M(Y_i) = Q_i[M(X_i)]$ с по-

мощью двух транзакций $M \xleftarrow{Y_i} Q_i \xleftarrow{X_i} M$. Обработка всех кубитных элементов на входном воздействии X дает возможность сформировать вектор M состояния цифровой системы (System State Vector – SSV), который при необходимости имеет и выходные переменные Y (рис. 6.4b) для управления другими компонентами вычислительной структуры. Быстродействие анализа (запись-считывание) векторной (линейной) структуры из $n = p \times q$

примитивов имеет оценку: $\gamma = (R + W) \times n$. Строго последовательный характер обработки упорядоченных по возрастанию номеров выходов квантовых примитивов можно усовершенствовать в сторону уменьшения времени анализа схемы. Для этого необходимо построить уже двумерную структуру – матрицу Q-элементов, ранжированных по уровням параллельной обработки групп примитивов, оформленных в столбцы (рис. 6.4c). Быстродействие такой структуры, по сравнению с линейной повышается в q раз, что становится уже соизмеримым со временем обработки комбинационной схемы на

$$\gamma = \frac{1}{q} (R + W) \times p$$

основе жестко заданных гальванических связей . Но глав-

ное преимущество матрицы «мягких» связей компонентов – наличие нижней строки запасных примитивов для ремонта в режиме online, что делает любой проект, содержащий комбинационную логику, тестопригодным, благодаря оперативной переадресации отказавшего Q- примитива на элемент из ремонтного запаса. Для этого желательно при синтезе такой двумерной структуры располагать в одном столбце однотипные примитивы, что уменьшит издержки на количество запасных элементов.

Инновационная идея матрицы кубитных примитивов характеризуется реализацией комбинационных Q-покрытий на адресуемых: координатах кубита и элементах памяти, мягко соединенных в цифровую схему с помощью вектора состояний линий, что дает возможность ремонтировать отказавшие логические примитивы в реальном времени с помощью их переадресации на запасные компоненты при достаточно высоком быстродействии функционирования вычислительного устройства. Платой за такие преимущества является существенная аппаратная избыточность по сравнению с жесткой комбинационной схемой, которая заключается в добавлении следующих компонентов (рис. 6.5): CU – устройство управления; Q-ADC – дешифратор адресов примитивов; M-SSV – вектор состояния цифровой системы; X-ADC – дешифратор адресов входов; X-M – память входов; Y-ADC – дешифратор адресов выходов; Y-M – память выходов.

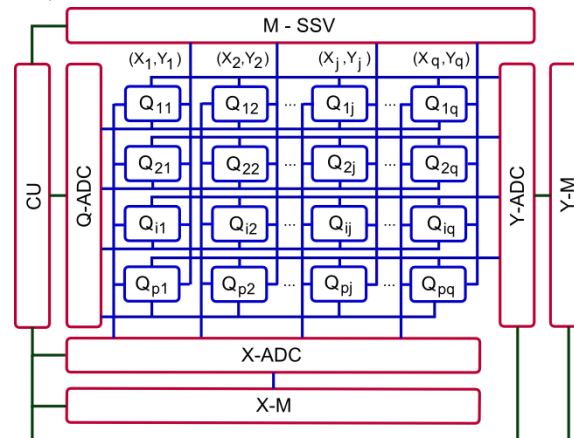


Рис. 6.5. Матричная структура операционного устройства

Упрощение автомата Мура до пары транзакционно взаимодействующих компонентов на основе M-вектора моделирования, адресно структурирующего элементы, привело к тривиальной модели MQT-автомата:

$$\begin{cases} A = \langle M, Q, X, Y \rangle, \\ M(Y_i) = Q_i[M(X_i)]. \end{cases}$$

Привязка идентификаторов функциональных Q-примитивов к номеру (адресу) выходной линии цифровой системы дает возможность упростить MQT-

автомат управления еще на одну транзакционную операцию, связанную с формированием адреса состояния выхода примитива Y :

$$\begin{cases} A = \langle M, Q, X \rangle, \\ M_i = Q_i[M(X_i)]. \end{cases}$$

Здесь уже нет системы выходных переменных, которые заменяются координатами M -вектора моделирования линий цифрового устройства. Представленный MQT-автомат имеет особенности: 1) Отсутствие жестких гальванических связей между элементами, формирующими комбинационную логику; 2) Наличие «мягких» или гибких связей Q -примитивов, формируемых за счет использования M -вектора (моделирования) состояния цифровой системы (System State Vector – SSV); 3) Частичное уменьшение быстродействия вычислительной структуры за счет устранения гальванических связей; 4) Обеспечение новой возможности комбинационной структуры, которая заключается в восстановлении работоспособности системы в случае возникновения отказов в примитивных элементах; 5) Инвариантность реализации MQT-компьютеров по отношению к среде использования (виртуальное пространство, критические объекты) и субстанции имплементации (твердые кристаллы и материалы, жидкие, газообразные и плазменные формы существования материи); 6) Структура адресного взаимодействия Q -компонентов памяти между собой посредством M -вектора моделирования (сосредоточение входов и выходов цифровой системы) напоминает технологию использования всех устройств компании Apple, когда входом и выходом любого гаджета является киберпространство (интернет или облака данной компании). Иначе, можно сделать вывод – как на макро, так и на микро уровне среды применения автоматов уже нет необходимости использовать внешние входы и выходы цифровой системы, которая уже сегодня ориентирована на жесткое взаимодействие с киберпространством и не представляет особого интереса как автономное вычислительное изделие. Вместо входов и выходов существует единое адресное пространство, в данном случае вектор M , откуда вычислительная система берет входные данные, а после преобразований она записывает туда же результаты или выходные данные:

$$\begin{cases} A = \langle M, Q \rangle, \\ M_i = Q_i(M_j). \end{cases}$$

Поэтому MQT-модель системного описания структур данных подобна отношениям между вычислительными устройствами (Q -компоненты) и интернетом (M -память). По-видимому, это есть настоящее и будущее технологической культуры, когда компьютер (гаджет) связывается с киберпространством, которое является входом и выходом для любого вычислительного устройства. 7) Что касается обработки кубитных вычислительных MQT-структур более низкого уровня, то здесь работает простая и эффективная транзакционная процедура – кубит функциональности берет данные из адресного пространства M , а затем туда же записывает результат преобразования: $M_i = Q_i(M_j)$. 8) Иначе, предлагаемая MQT-модель вычислительной

ячейки киберпространства включает структурный MQ-автомат адресной мягкой организации взаимодействия функциональных примитивов $MQ = \{M, Q, [(M \square Q) \rightarrow M] \square [M_Y = Q(M_X)]\}$ посредством M-вектора моделирования и MT-автомат управления обработкой кубитных примитивов на основе использования единственной операции транзакции, регламентируемой характеристическим выражением $M_i = Q_i(M_j)$, определяющим взаимосвязи адресуемых компонентов и процедуру их транзакционной обработки. Таким образом, предложенный MQT-автомат, используя только элементы памяти и единственную операцию транзакции, дает возможность системно проектировать высоконадежные вычислительные и информационные сервисы как в реальном, так и в виртуальном мире.

В качестве примера далее предлагается анализ цифровой схемы на одном тестовом наборе, для которой вектор моделирования и кубитная структура представлены ниже:

M =		1	2	3	4	5	6	7	8	9	A	B	C
		0	0	0	1	1	1	0	0	1	0	1	0
Q _{ij}		1			2			3					
1	1	0001A			A			8			1110B		
	2				7			6					
2	3	00017			7			6			1110C		
	4				5			9					

Здесь сущность анализа цифровой структуры заключается в заполнении вектора моделирования по всем координатам: 1) Запись в вектор моделирования M по шести координатам входных переменных двоичного набора 000111. 2)

Обработка примитива с номером Q_{11} , который имеет входные переменные 1 и 2, а выход, обозначенный символом A. Для этого формируется адрес 00 путем конкатенации содержимого ячеек 1 и 2 вектора M. Применяя данный адрес к Q-вектору 0111, определяется содержимое нулевой ячейки, равное нулю, которое записывается в вектор M по адресу A выходной переменной обрабатываемого кубита. 3) Повторение процедуры анализа, описанной в пункте 2, ко всем Q-примитивам дает возможность полностью определить двоичными сигналами координаты вектора моделирования: 000111001010.

Аппаратная имплементация приведенного и масштабируемого примера кубитной структуры цифрового устройства на основе использования элементов памяти, представлена на рис. 6.6.

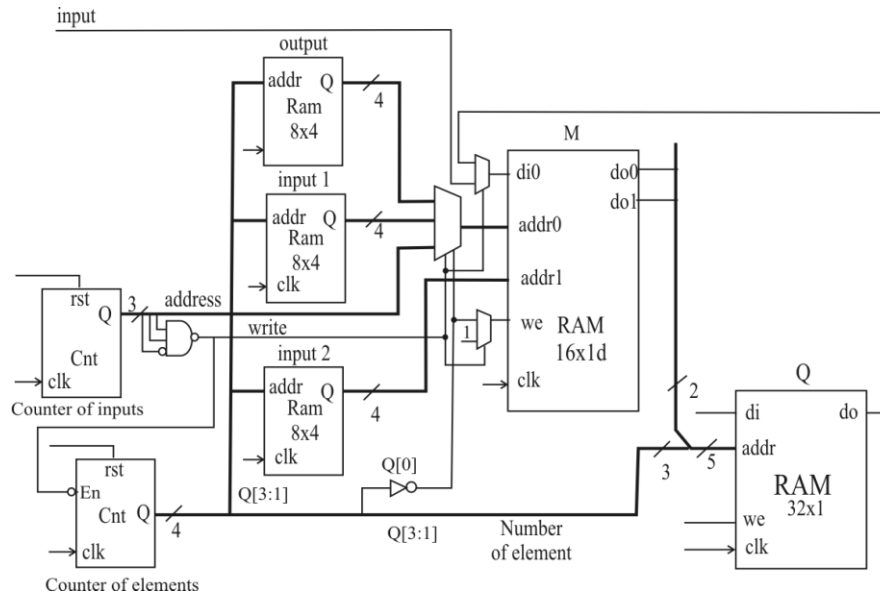


Рис. 6.6. Аппаратная реализация кубитной структуры комбинационной схемы

Структура схемной реализации содержит переменные и функциональные элементы, имеющие следующее назначение: *input* – вход для последовательного занесения входных значений вектора *M*; *rst* – общий сброс системы, в данном случае счетчиков; *clk* – вход синхронизации; *counter of inputs* – счетчик заполнения входных координат вектора *M*; *counter of element* – счетчик номера обрабатываемого примитива, который предоставляет два такта для считывания входного набора из двух координат вектора *M*; *Q[3:1]* – шина номера обрабатываемого примитива; *Q[0]* – переменная режима считывания входных значений из вектора *M* или записи результата в *M*. Блоки памяти: *Ram 8x4 output* – хранит номера выходных линий примитивов; *Ram 8x4 input 1* и *Ram 8x4 input 2* – сохраняют номера входных линий примитивов. *Ram 16x1d* – двухпортовая память хранения вектора моделирования *M*, где *addr0* – адрес входа 1 при значении 00 на входах управления мультиплексора, адрес записи результата при значении 01 на входах управления мультиплексора, адрес для инициализации входных данных при значении 1X на входах управления мультиплексора; *addr1* – адрес входа 2 обрабатываемого примитива; *di0* – вход данных памяти при обработке примитива (*MUX=1*) или внешний вход *input* при инициализации входных данных (*MUX=0*); *we* – разрешение записи в вектор *M*; *do0* – выход, соответствующий входу *addr0*; *do1* – выход, соответствующий входу *addr1*. Компонент *RAM 32x1* предназначен для хранения *Q*-векторов задания функциональностей комбинационной схемы: *di* – вход данных, может быть использован для инициализации (записи) структуры кубитов; *addr* – [4:0]; *addr[4:2]* – номер элемента,

addr[1:0] – входной набор для примитива. Сложность аппаратной реализации примера комбинационной схемы составляет 150 вентилей, включающих 20 LUT системы элементов компании Xilinx Spartan 3E. Быстродействие функционирования или формирования вектора моделирования равно 180 ns.

6.1.4. MQT–структуры для будущих исследований

Введение MQT–автомата для синтеза и анализа вычислительных сервисов позволяет сделать и более общие выводы. Мир имеет дуализм материально-информационной целостности, а значит детерминированной воспроизводимости объектов, процессов и явлений. Любой материальный объект способен хранить информацию, следовательно, его можно использовать для записи и считывания, а значит – для создания вычислительной системы, пусть даже в некоторой степени специфической. Человек есть биологический компьютер с заданной программой функционирования (жизнедеятельности) всех его органов во времени. Как любую программу, код (геном) функционирования человека можно считать-записать, а значит – можно и скорректировать. Любые объекты на планете имеют свои программы существования во времени – информационные геномы. Процессы и явления на планете также имеют свои программы развития, но более высокого уровня иерархии в управлении. Вселенная также имеет собственный геном – информационную программу развития. Естественно, что любую программу можно считать-записать, а значит скорректировать – это оптимистическая нота информационной модели объектов, процессов и явлений материального мира! Декодировать программу, как и дизассемблировать двоичный код, сложно, но возможно. Для этого нужно определить место – носитель кода и способ шифрования каждого объекта, процесса или явления природы. Можно предположить, что геном биологического объекта имеет форму, подобную структуре многозначных кубит-векторов, когда каждый из них

$$Q = (q_1, q_2, \dots, q_i, \dots, q_n), q_i \in \{a_1, a_2, \dots, a_j, \dots, a_k\}$$

может быть трансформирован в двоичную матрицу, где символ алфавита описания разрядов кванта может быть развернут в двоичный вектор или код-столбец. Многозначность функциональности достаточно распространена в природе, удобна для восприятия глазом человека, но для компьютера символы любого алфавита пока трансформируются в двоичные векторы или коды. Геном человека с позиции компьютерной инженерии есть структура данных, записанная в биопамять, компактная программа развития и существования во времени и пространстве белковых конструктивов, которая имеет аппарат управления-мониторинга (чтения-записи) без существования жестких гальванических связей. Такие принципы биоинженерии следует применять и к объектам киберпространства, чтобы обеспечить их надежность, ремонтпригодность, восстанавливаемость на микро и макроуровнях.

Таким образом, кубитные структуры дают возможность сделать из комбинационной схемы простейший автомат (интегрирующая память, кванты функ-

циональностей, операция транзакции) и перейти от программного моделирования цифровых систем к аппаратной эмуляции структур и процессов, составляющей уже пользовательские функции компьютера, инвариантные по отношению к технологиям имплементации. Показательным аналогом и прообразом служит аппаратный ускоритель процессов моделирования PRUS [7] доктора Stanley Hyduke, Aldec Inc., ориентированный на уменьшение времени проектирования и верификации цифровых систем на кристаллах. Но в сравнении с [7] здесь предлагается использовать по прямому функциональному назначению процессор «мягкого» адресно ориентированного аппаратного моделирования кубитных структур в качестве вычислительного изделия, доставляющего сервисы потребителю. При этом сохраняется высокое быстродействие функционирования устройства, дополненное существенной для критических систем возможностью восстановления работоспособности в режиме реального времени.

Технологическая сингулярность (Вернор Виндж) – взрыв в понимании законов вселенной на коротком промежутке времени и создание кибермозга человечества в целях повышения качества жизни очевидно произойдет благодаря развитию трех компонентов: биоинженерии, искусственного интеллекта киберпространства и нанотехнологий проектирования. Здесь имеется ввиду: 1) встроенная и непосредственная интеграция человека с компьютером (киберпространством), путем устранения языковых барьеров и интерфейсов между ними; 2) создание искусственного интеллекта для самообучения, во-вторых, и самосовершенствования, во-первых, когда компьютеры и киберпространство будут способны к самовоспроизводству более совершенных моделей, структур и процессов; 3) «нано-выращивание» компьютера путем структурирования атомов. Одним из главных условий достижения данной точки развития технологической культуры является создание принципиально нового компьютера (вычислительного процесса) инвариантного по отношению к любой точке пространства или его субстанции, которую можно структурировать и использовать вместо кремния. Пространственный параллелизм существующих компьютеров и точечная одновременная многозначность – два пути в развитии интеллектуального киберпространства. Иначе, компьютер можно сделать на чем угодно, что способно сохранять во времени и/или в пространстве не менее двух управляемых и мониторируемых состояний. Регистр или память, как основа для хранения данных и выполнения вычислительных процедур, представляет собой несколько физических точек или триггеров, к которым можно применить параллельную операцию, разнесенную в пространстве. При этом повышение производительности цифрового устройства сегодня осуществляется только за счет расширения субстанции вычислительной структуры. Вопрос – можно ли объединить упомянутые точки пространства в одну для выполнения параллельных операций над совокупностью данных, сосредоточенных в одной материальной точке? Возможно так и ставился практическими учеными вопрос, который предсказал появление квантового компьютера. Несомненно, что будущее компьютерной индустрии должно быть связано с многовариантностью физической точки пространства, как примитива структур данных, которая должна иметь воз-

возможность хранения булеана состояний в качестве операнда для реализации вычислительных процедур. Но здесь возникает только одно решение, заслуживающее внимания. Все, что физически является стабильным во времени, может иметь только одно состояние в каждый конкретный момент времени, включая цифровые дискретные автоматы, устройства и системы. Решение совершенно очевидно – дуализм частицы и волны на всем спектре электромагнитного излучения обладает искомыми свойствами пространственной точки: 1) Каждая точка пространства может иметь в конкретный момент времени множество волн или квантов из известных и неизвестных человеку диапазонов; 2) Это означает возможность наличия в точке пространства множества частиц соответствующих диапазонов, которые формируют совокупность информационных состояний данной точки; 3) Неопределенность (многовариантность) во времени процессов и явлений есть ключ к созданию принципиально новых компьютеров, которые можно называть как квантовыми, так и электромагнитными, учитывая дуализм электродинамики в отношении носителя (частица или волна); 4) На каждой стадии развития технологической культуры необходимо выбирать подходящую точку пространства или субстанции (газообразная, жидкая, твердая, плазменная), которую можно использовать для мониторинга и управления ее многозначных состояний с помощью существующих технологий. Предсказания фантаста Станислава Лема и ученого В.И. Вернадского о возможности существования разума в жидкой (океан Солярис) и газообразной среде (ноосфера) сегодня уже близки во времени к созданию глобального искусственного разума киберпространства, который мы должны получить к 2050 году; 5) Дискретизация спектра частот в каждой точке пространства может служить основой для кодирования как примитивов (двоичных, многозначных), так и более сложных ассоциаций, например, булеана состояний. Иначе, 8-разрядный регистр, имеющий 256 состояний, можно представить одной точкой в пространстве для последующей обработки на основе мониторинга и управления соответствующими квантами или спектром частот. 6) Если предположить, что кванты существуют во всех диапазонах непрерывного и бесконечного спектра, то вопрос заключается в том, какой диапазон частот в точке пространства или субстанции будет освоен первым для создания новых как мощных, так и простых квантовых электродинамических вычислителей на основе экономически приемлемых технологий мониторинга и управления (атомными примитивами) частицами или волнами. 7) Рыночная привлекательность «зеленых» нанотехнологий («восходящее» проектирование) – построение или выращивание вычислителя (квантового компьютера) путем структурирования атомов – заключается в безотходности, микроминиатюрности, сверхнизком энергопотреблении, абсолютно минимальной затратности материалов, а в будущем и стоимости, сверхвысоком быстродействии и требуемой (по принципу разумной достаточности) масштабируемости, соизмеримой с классом доставляемых сервисов. Микротехнологии («нисходящее» проектирование) – суть которых: взять заготовку и отсечь все лишнее, становятся все менее привлекательными для рынка, ввиду несовершенства качества и сверхзатратности создаваемых изделий.

Ученые научились сегодня не только сканировать атомные структуры, но и последовательно строить или выращивать их. Однако на пути решения проблемы нано-технологического направления разработки квантового компьютера на рынке имеется три привлекательных пункта: 1) Открытие технологий с высоким быстродействием выращивания требуемых гетерогенных атомных структур в соответствии с заданной программной спецификацией вычислителя; 2) Создание эффективного транзакционного механизма для реализации простого мониторинга и управления квантовыми состояниями выращенных атомных вычислительных структур с адресуемыми компонентами; 3) Обеспечение требуемой стабильности во времени состояний компонентов атомной структуры, реализующей память; 4) Для решения проблемы необходимо объединить достижения и усилия ученых, работающих в областях: квантовой физики атомных взаимодействий, биоинженерии, микроэлектроники, электрохимии и компьютерной инженерии.

6.1.5. Выводы

1. В данном разделе предложены кубитные модели описания цифровых систем и компонентов, которые характеризуются компактностью описания таблиц истинности в форме Q-покрытий благодаря унитарному кодированию входных состояний, что дает возможность повысить быстродействие программных и аппаратных средств интерпретативного моделирования вычислительных устройств за счет адресной реализации анализа логических примитивов.
2. Представлена матричная модель кубитных примитивов для реализации комбинационных схем, которая характеризуется адресным объединением Q-покрытий на элементах памяти, «мягко» соединенных в цифровую схему с помощью вектора состояний линий, что дает возможность ремонтировать отказавшие логические примитивы в реальном времени с помощью их переадресации на запасные компоненты при достаточно высоком быстродействии функционирования вычислительного устройства.
3. Введена гибкая автоматная MQT-модель компьютера, которая характеризуется использованием только адресуемых структур памяти и операции транзакции для программной и аппаратной реализации комбинационных и последовательностных функциональностей, что дает возможность создавать быстродействующие и надежные вычислители для создания сервисов киберпространства на основе параллельных логических операций и ремонта неисправных адресуемых функциональных примитивов.
4. Описана инновационная идея квантовых вычислений, которая характеризуется переходом от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение (точку) к логическим регистровым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений, что дает возможность определить новые перспективы на пути создания высокопроизводительных компьютеров параллельного анализа и синтеза структур и сервисов дискретного киберпространства.

5. Представлена концепция системного проектирования программных и аппаратных облачных сервисов киберпространства на основе MQT-автомата, которая характеризуется применением транзакций на кубитных адресно связанных компонентах вычислителя с помощью вектора состояний (моделирования), что дает возможность программировать логические функциональности в кубитных примитивах интерпретативной структуры цифрового устройства.

6. Рыночная привлекательность MQT-компьютера определяется: примитивизмом его реализации в программном и аппаратном исполнении; высоким уровнем использования памяти в структуре современных вычислителей, достигающим 94% от площади кристалла; и как следствие – уменьшением влияния комбинационной логики на быстродействие системы в целом; повышением надежности компьютеров за счет онлайн-ремонта адресуемых элементов памяти, включая логические примитивы. Существующее неприятие MQT-компьютера рынком связано с лоббированием вычислителей на основе жесткой комбинационной логики в VLSI-проектах со стороны компаний, разрабатывающих процессоры на основе технологий reusable logic.

7. Направления будущих исследований: Синтез кубитных моделей цифровых систем на основе использования Q-покрытий, а также разложение (анализ) функциональностей на кубитные компоненты. Моделирование неисправностей и синтез тестов на основе использования Q-покрытий компонентов цифровых систем. Разработка специализированных программных приложений для решения задач синтеза комбинационных устройств на основе использования матричной модели MQT-процессора, ориентированного на высокий параллелизм решения практических задач и восстановление работоспособности устройства в реальном времени.

6.2. Кубитные технологии анализа и диагностирования цифровых устройств

6.2.1. Эволюция киберпространства планеты. Развитие киберпространства условно делится на следующие периоды: 1) 1980-е годы – формирование парка персональных компьютеров; 2) 1990-е годы – внедрение Интернет-технологий в производственные процессы и быт человека; 3) 2000-е годы – повышение качества жизни за счет внедрения мобильных устройств и облачных сервисов; 4) 2010-е годы – создание цифровой инфраструктуры мониторинга, управления и взаимодействия между собой стационарных и движущихся объектов (воздушный, морской, наземный транспорт и роботы); 5) 2015-е годы – создание глобальной цифровой инфраструктуры киберпространства, где все процессы, явления идентифицируются во времени и в трехмерном пространстве и становятся интеллектуальными. В связи с необходимостью развития параллельных вычислений для неупорядоченных данных в последние годы становятся все более значимыми кубитные структуры для создания облачных Интернет сервисов, благодаря их позитивной альтернативности существующим затратным по времени классическим моделям

последовательной обработки теоретико-множественных структур за счет существенного расширения памяти [23]. Однако такая плата в настоящее время вполне допустима, поскольку рынок nano-электронных технологий предоставляет сегодня разработчикам цифровых систем до 1 миллиарда вентилей на кристалле размерностью 2x2 см при толщине пластины в 5 микрон. При этом современные технологии допускают создание пакета или «сэндвича», содержащего до 7 кристаллов. Практически «беспроводное» соединение таких пластин основывается на технологической возможности сверления порядка 10 тысяч сквозных отверстий (vias) на 1 квадратном сантиметре. Кроме того, появление трехмерных FinFETs транзисторов и основанных на них 3D-технологий реализации объемных цифровых систем предоставляют новые возможности для создания более быстродействующих за счет уменьшения задержек параллельных вычислительных устройств [29-33, 36]. Поэтому можно и нужно использовать «жадные» к аппаратуре модели и методы для создания быстродействующих средств параллельного решения практических задач. Имея в виду дискретность и многозначность алфавитов описания информационных процессов, свойство параллелизма (одновременности процессов), заложенное в квантовых вычислениях, является востребованным при создании эффективных и интеллектуальных «движков» для киберпространства, облачных структур и сервисов Интернета; повышения надежности цифровых устройств; тестирования и моделирования дискретных систем на кристаллах. Здесь не рассматриваются физические основы квантовой механики, касающиеся недетерминированного взаимодействий атомных частиц, но используется понятие кубитной структуры как векторной формы совместного или одновременного задания булеана состояний в конечной и дискретной области киберпространства, ориентированного на параллелизм и суперпозицию обработки предлагаемых кубитных моделей и методов.

Квантовые эмуляторы на классических компьютерах достаточно эффективно применяются для решения оптимизационных задач, связанных с полным перебором вариантов решений на основе использования теории множеств [23, 27]. Особенность в том, что множество элементов в компьютере всегда является упорядоченным, поскольку каждый бит, байт или другой компонент имеет свой адрес. Поэтому все теоретико-множественные операции, так или иначе, сводятся к полному перебору адресов примитивных элементов. Адресный порядок структур данных хорош для задач, где компоненты моделей можно строго ранжировать, что дает возможность выполнять их анализ за один проход или одну итерацию. Там, где нет порядка в структуре, например, множество всех подмножеств, классическая модель памяти и вычислительных процессов наносит вред времени анализа ассоциации равных по рангу примитивов, или, в лучшем случае, обработка ассоциативных групп является неэффективной. Что можно предложить для неупорядоченных данных вместо строгого порядка? Процессор, где элементарной ячейкой служит

образ или шаблон универсума из n примитивов, который генерирует $Q = 2^n$ всех возможных состояний такой ячейки в виде булеана или множества всех подмножеств. Прямое решение, ориентированное на создание такой ячейки

использует унитарное позиционное кодирование состояний примитивов, которое с помощью суперпозиции последних образует универсум примитивов, формирующее в пределе – булеан или множество всех подмножеств [27, 37]. n-Кубит есть векторная форма унитарного кодирования универсума из n

примитивов для задания булеана состояний 2^{2^n} с помощью 2^n двоичных переменных. Например, если $n=2$, то 2-кубит задает 16 состояний с помощью 4-х переменных. Если $n=1$, то кубит задает 4 состояния на универсуме из двух примитивов с помощью 2-х двоичных переменных (00,01,10,11) [23]. При этом допускается суперпозиция (одновременное существование) в векторе 2^n состояний. Кубит (n-кубит) дает возможность использовать логические операции вместо теоретико-множественных для существенного ускорения процессов синтеза и анализа дискретных систем. Далее кубит отождествляется с n-кубитом или вектором, если это не мешает пониманию излагаемого материала. Поскольку квантовые вычисления связаны с анализом кубитных структур данных, то далее частично эксплуатируется определение «квантовый» для идентификации технологий, использующих два свойства квантовой механики: параллелизм обработки и суперпозицию состояний.

6.2.2. Кубитный метод диагностирования цифровых систем. Предлагается метод диагностирования функциональных нарушений и константных неисправностей в программных или аппаратных блоках, которые используют «кубитные» или многозначные структуры данных для задания диагностической информации, что дает возможность существенно уменьшить вычислительную сложность процессов моделирования и диагностирования за счет введения параллельных логических операций над матричными данными. Представлен кубитный метод исправного моделирования цифровых устройств с восстановлением работоспособности компонентов цифровой системы в режиме online, который имеет существенно более высокое быстродействие за счет адресной реализации процедуры обработки функциональных примитивов, заданных кубитными векторами состояний выходов.

Модель объекта диагностирования представлена в форме графа цифровой системы, которая имеет функциональные элементы, соединенные линиями связей. Среди них имеются ассерции – точки наблюдения или мониторинга, необходимые для верификации, тестирования и диагностирования неисправностей [30]. Диагностическая информация представлена компонентами:

1) Тест проверки или диагностирования неисправностей заданного класса, в данном случае рассматриваются константные дефекты $\{=0,=1\}$ линий схемы. 2) Таблица неисправностей [31], строки которой задают векторы проверяемых на каждом тестовом наборе дефектов, соответствующих линиям схемы. 3) Матрица достижимостей, которая определяет достижимость каждой ассерционной точки – наблюдаемой линии – со стороны множества предшествующих линий схемы [27]. 4) Матрица состояния ассерционного механизма или матрица экспериментальной проверки, задающая состояние

каждой ассерции на тестовых наборах путем сравнения эталонной реакции в данной точке с реальным сигналом в процессе выполнения диагностического эксперимента [30,36].

Базовая модель диагностирования цифрового изделия, дискретного процесса или явления представлена компонентами, которые создают 4 измерения в пространстве признаков:

$$\begin{cases} D_b = \langle S, A, F, T \rangle \\ D = \{ \langle S, A \rangle, \langle F, T \rangle \}; \\ \begin{cases} V_b = (|S| \times |A| \times |F| \times |T|); \\ V = (|S| \times |A|) + (|F| \times |T|); \end{cases} \\ V_b \gg V; \\ \begin{cases} S^* = f(S, A, T); \\ A^* = g(T, A); \\ F^* = h(S, A, F, T); \end{cases} \end{cases}$$

При этом в модели объем диагностической информации V формируется декартовым произведением (мощностей) четырех компонентов в порядке следования, указанном выше: 1) структура объекта; 2) механизм ассерций или мониторинга; 3) совокупность неисправностей или модулей, подверженных функциональным нарушениям; 4) тестовые наборы или сегменты для диагностирования неисправностей или совокупности упомянутых модулей. Актуально и существенно уменьшить объем диагностической информации можно путем понижения размерности пространства признаков за счет разделения базовой модели на два непересекающихся подмножества $\langle S, A \rangle, \langle F, T \rangle$. В этом случае оценка объема диагностической информации становится не мультипликативной, а аддитивной по отношению к мощности полученных в результате разбиения подмножеств без какого-либо уменьшения глубины диагностирования. Здесь первый компонент модели диагностирования представлен матрицей достижимостей, которая позволяет минимизировать маску возможных дефектов на основе анализа структуры схемы путем сравнения истинных и реальных результатов моделирования выходных сигналов на каждом тестовом наборе или сегменте. Число строк такой матрицы равно количеству наблюдаемых выходов или ассерций.

В процессе выполнения метода диагностирования создается двоичная матрица структурной активизации неисправностей, которая служит маской для существенного уменьшения множества подозреваемых дефектов при совместном анализе таблицы неисправностей. При этом символы одиночных константных дефектов $\{0, 1, X, \emptyset\}$, $X = \{0, 1\}$ в ячейках таблицы неисправностей [31] кодируются соответствующими состояниями кубита (10,01,11,00)

многозначного алфавита Кантора $A^k = \{0, 1, X, \emptyset\}$, что дает возможность исключить из вычислительных процессов теоретико-множественные процедуры, заменив их на векторные логические операции.

Для рассмотрения сущности предлагаемого метода используется фрагмент цифровой схемы, представленной на рис. 6.2.1. Здесь имеются три ассерционных точки A, B, C для наблюдения за состоянием всех линий схемы в процессе тестирования (выполнения диагностического эксперимента) путем подачи пяти тестовых воздействий, заданных в таблице неисправностей F(T). Координаты данной таблицы задают проверяемые на тест-векторах неисправности 0 и 1, а также имеются состояния координат: $\emptyset$ (.) – отсутствие проверяемых дефектов и X – проверка на линии константы 0 и 1 одновременно. Правая часть таблицы есть матрица состояний ассерционного механизма в виде результатов сравнения эталонной и реальной реакций цифрового устройства на тестовые наборы. Значение 1 означает несравнение, 0 – совпадение упомянутых реакций.

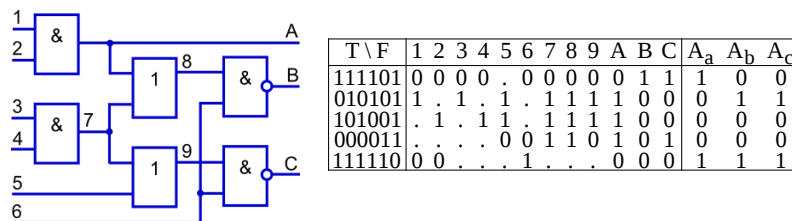


Рис. 6.2.1. Фрагмент цифровой схемы и таблица неисправностей

В таблице неисправностей не учитывается структура схемы для повышения глубины диагностирования на основе вычисления реальной матрицы состояний ассерционного механизма, которая совместно с матрицей достижимостей создает структурную маску, минимизирующую множество подозреваемых дефектов. Для фрагмента цифровой схемы, представленной на рис. 1, матрица достижимостей имеет следующий вид:

S = S _{ij}	1	2	3	4	5	6	7	8	9	A	B	C
1	1	1	.	.	.	.	.	.	1	.	.	.
2	1	1	1	1	.	1	1	1	.	1	.	.
3	.	.	1	1	1	1	1	.	1	.	.	1

Здесь выходы-ассерции A, B, C являются мониторами технического состояния объекта диагностирования. Каждый из них может иметь два значения: $A_{ij} = \{0,1\}$, которые определяют матрицу экспериментальной проверки

$A = |A_{ij}|$ путем сравнения эталонных $T = |T_{ij}|$ и реальных $U = |U_{ij}|$ состояний наблюдаемых или выходных линий: $A_{ij} = T_{ij} \oplus U_{ij}$, которые формируют маску возможных дефектов с помощью следующего выражения:

$S_i = S(T_i) = \left(\bigvee_{A_{ij}=1} S_{ij} \right) \wedge \left(\bigvee_{A_{ij}=0} S_{ij} \right)$. Каждый тест-вектор (-сегмент) активизи-

рует собственную структуру возможных дефектов, которая функционально зависит от маски, ассерций (состояния наблюдаемых выходов) и тестовых

наборов: $S = f(S, A, T_i)$. Если предположить, что в матрице $S = |S_{ij}|$ состояния асерционных выходов на первом тест-векторе равны $A_{1A} = 0; A_{1B} = 1; A_{1C} = 1$, где значение 1 идентифицирует проявление дефекта в устройстве, то маска возможных дефектов, согласно функционалу $S_1 = S(T_1) = (\bigvee_{A_{1j}=1} S_{1j}) \wedge (\overline{\bigvee_{A_{1j}=0} S_{1j}})$, будет иметь следующий вид:

$$S_1 = S(T_1) = (S_2 \vee S_3) \wedge (\overline{S_1}) = (111101110010 \vee 0011111101001) \wedge \overline{(110000000100)} = (111111111011) \wedge (001111111011) = (001111111011).$$

Полученная маска накладывается на первую строку таблицы неисправностей, что определяет множество подозреваемых дефектов $F_1 = T_1 \wedge S_1 \rightarrow F_1 = T_1 \wedge S_1$, формирующих асерционную выходную реакцию $A_{1(A,B,C)} = (011)$ устройства на первый тест-вектор:

Faults	1	2	3	4	5	6	7	8	9	A	B	C
T_1	0	0	0	0	.	0	0	0	0	0	1	1
S_1	0	0	1	1	1	1	1	1	1	0	1	1
$F_1 = T_1 \wedge S_1$	.	.	0	0	.	.	0	0	0	.	1	1

В соответствии с предложенной процедурой получения маски одной строки выполняется построение матрицы структурной активизации неисправностей $S(T)$ на основе использования таблицы экспериментальной проверки $A = |A_{ij}|$, задающей состояния асерционного механизма в процессе выполнения тестирования $S(T) = S \otimes A$:

$S = S_{ij} $	1	2	3	4	5	6	7	8	9	A	B	C
1	1	1	.	.	.	.	.	.	1	.	.	.
2	1	1	1	1	.	1	1	1	.	1	.	.
3	.	.	1	1	1	1	1	.	1	.	.	1

 $\otimes$

$A = A_{ij} $	A	B	C
T_1	1	0	0
T_2	0	1	1
T_3	0	0	0
T_4	0	0	0
T_5	1	1	1

 $=$

$S(T)$	1	2	3	4	5	6	7	8	9	A	B	C
T_1	1	1	0	0	0	0	0	0	0	1	0	0
T_2	0	0	1	1	1	1	1	1	0	1	1	1
T_3	0	0	0	0	0	0	0	0	0	0	0	0
T_4	0	0	0	0	0	0	0	0	0	0	0	0
T_5	1	1	1	1	1	1	1	1	1	1	1	1

В целях формирования структур данных, удобных для компьютерной обработки, необходимо перевести символы таблицы неисправностей в двухразрядные коды в соответствии с правилами $\triangleright$ -кодирования: $\triangleright = \{0=10; 1=01, X=11, \emptyset=00\}$, применение которых к таблице неисправностей $F(T)$ дает следующий результат:

$F(T)$	1	2	3	4	5	6	7	8	9	A	B	C
T_1	00	00	00	.	00	00	00	00	01	11	.	.
T_2	1.	1.	1.	1.	11	11	11	00	00	01	01	01
T_3	.	1.	11	1.	11	11	11	00	00	00	01	10
T_4	.	.	.	.	00	01	10	10	01	01	01	01
T_5	00	00	.	.	1.	.	.	00	00	00	00	00

 $\triangleright$

$F(T)$	1	2	3	4	5	6	7	8	9	A	B	C
T_1	10	10	10	10	00	10	10	10	10	01	01	01
T_2	01	00	01	00	01	00	01	01	01	01	10	10
T_3	00	01	00	01	01	00	01	01	01	01	10	10
T_4	00	00	00	00	10	10	01	01	10	01	10	01
T_5	10	10	00	00	00	01	00	00	00	10	10	10

После получения структурной матрицы $S(T)$, предназначенной маскировать реальные дефекты в таблице неисправностей и ее кодированной формы, необходимо выполнить $\#$ -суперпозицию двух матриц: $F(T) = S(T) \# F(T)$, которая сводится к выполнению $\#$ -операции над одноименными координатами $F_{ij} = \bar{F}_j \leftarrow (F_j = 00) \vee (S_{ij} = 0)$, что означает модификацию кодов координат таблицы $F(T)$ при выполнении заданных условий. Иначе, данная операция сводится к инверсии ячеек матрицы кодов неисправностей, маскируемых нулевыми сигналами структурной матрицы активизации, а также всех нулевых кодов таблицы неисправностей. Таблица истинности данной $\#$ -операции в символьном и кодированном виде представлена ниже:

$\# = S_{ij} \setminus F_{ij}$	$\emptyset$	1	0	X
0	X	0	1	$\emptyset$
1	X	1	0	X

$\# = S_{ij} \setminus F_{ij}$	00	01	10	11
0	11	10	01	00
1	11	01	10	11

Таблица истинности скорректирована относительно инверсии состояния 00 в 11 при единичном значении сигнала активизации неисправности, потому что такой код (00) означает присутствие в схеме на линии пустого множества проверяемых дефектов, что невозможно. Но код 00 еще блокирует все вычисления конъюнкции по столбцу, превращая результат в 00. Инверсия кода дает возможность не маскировать при логическом умножении действительно присутствующих дефектов, любых знаков. При этом предполагается, что невозможно тест-вектором проверить на одной линии схемы дефекты разных знаков.

Выполнение процедуры суперпозиции структурной матрицы с кодированной таблицей неисправностей $F(T) = S(T) \# F(T)$ дает следующий результат:

$S(T)$	1	2	3	4	5	6	7	8	9	A	B	C
T1	1	1	0	0	0	0	0	0	1	0	0	0
T2	0	0	1	1	1	1	1	1	0	1	1	1
T3	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0
T5	1	1	1	1	1	1	1	1	1	1	1	1

 $\#$

$F(T)$	1	2	3	4	5	6	7	8	9	A	B	C
T1	10	10	10	10	00	10	10	10	10	10	01	01
T2	01	00	01	00	01	00	01	01	01	01	10	10
T3	00	01	00	01	01	00	01	01	01	01	10	10
T4	00	00	00	00	10	10	01	01	10	01	10	01
T5	10	10	00	00	00	01	00	00	00	10	10	10

 $=$

$F(T)$	1	2	3	4	5	6	7	8	9	A	B	C
$\wedge$	10	10	01	01	11	01	01	01	01	10	10	10
	10	11	01	11	01	11	01	01	01	10	10	10
	11	10	11	10	10	11	10	10	10	10	01	01
	11	11	11	11	01	01	10	10	01	10	01	10
	10	10	11	11	11	01	11	11	11	10	10	10
$F(T) = \bigwedge_{i=1}^n F_i$	10	10	01	00	00	01	00	00	00	10	00	00
$F =$	0	0	1	.	.	1	.	.	.	0	.	.

На заключительной стадии диагностирования выполняется единственная и векторная операция логического умножения всех строк кодированной модифицированной таблицы истинности $F(T)$:

$$F(T) = \left(\bigvee_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \overline{F_i} \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \overline{F_i} \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigwedge_{A_i=0} \overline{F_i} \right) = \left(\bigwedge_{i=1}^n F_i \right).$$

Это дает возможность точно определить все дефекты, присутствующие в объекте диагностирования, которые представлены в двух нижних строках приведенной выше кодированной таблицы неисправностей $F(T)$:
 $F = \{1^0, 2^0, 3^1, 6^1, A^0\}$.

Теоретическое доказательство матричного диагностирования одиночных и кратных дефектов представлено ниже в виде двух теорем.

Теорема 6.1: Одиночные константные дефекты цифровой схемы, заданные кубитами на тестовых наборах многозначной таблицы неисправностей, определяются с помощью векторной and-операции, маскируемой по строкам матрицей (вектором) экспериментальной проверки $A = |A_{ij}|$ всех ассерционных точек:

$$F(T) = \left(\bigvee_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \overline{F_i} \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \overline{F_i} \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigwedge_{A_i=0} \overline{F_i} \right) = \left(\bigwedge_{i=1}^n F_i \right).$$

Выражение является верным, потому что: 1) Второй сомножитель – чистая математика – отрицание дизъюнкции есть конъюнкция отрицаний, что означает умножение кодов таблицы с их предварительным отрицанием. 2) Первый сомножитель ориентирован на поиск непротиворечивых дефектов, поэтому он заменяется на $\left(\bigwedge_{A_i=1} F_i \right)$. Действительно, на одной линии или пере-

менной не могут присутствовать одновременно две противоположных по знаку проверяемых неисправности. Поэтому в базовой формуле дизъюнкция дефектов $\left(\bigvee_{A_i=1} F_i \right)$ в большей степени ориентирована на поиск кратных не-

исправностей, но не связанных с одной линией. Кратность противоречивых дефектов на одной линии, равно как и инверсия пустого множества неисправностей, теоретически создает условия беспрепятственного умножения других ячеек столбца в целях формирования на каждой линии результата в виде дефекта одного знака или пустого множества неисправностей.

Теорема 6.2: Кратные константные дефекты цифровой схемы, заданные кубитами на тестовых наборах многозначной таблицы неисправностей, определяются с помощью векторных or- and- операций, маскируемых по строкам вектором экспериментальной проверки $A(T)$ всех ассерционных точек:

$$F(T) = \left(\bigvee_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \overline{F_i} \right) = \left(\bigvee_{A_i=1} F_i \right) \wedge \left(\bigwedge_{A_i=0} \overline{F_i} \right).$$

Выражение является верным, потому что: 1) Второй сомножитель есть отрицание дизъюнкции или конъюнкция отрицаний, что означает умножение кодов таблицы с их предварительным отрицанием. 2) Первый сомножитель ориентирован на поиск кратных дефектов в предположении, что на одной линии или переменной могут присутствовать одновременно две противополо-

ложных по знаку проверяемых неисправности. Данная формула в большей степени ориентирована на поиск кратных дефектов в блоках цифровых систем, не связанных с одной линией. Кратность неисправностей в цифровой системе теоретически создает условия для логического сложения других ячеек столбца в целях формирования результата в виде множества дефектов, формирующих заданный вектор экспериментальной проверки, из которых необходимо вычесть проверяемые на тесте неисправности, которые не оказывают влияния на формирование некорректных реакций по выходам.

Интерес представляет поиск кратных дефектов на основе мультипроцессора Хассе [29, 32], который ориентирован на решение задачи покрытия путем полного перебора событий, обеспечивающих точное покрытие вектора экспериментальной проверки столбцами таблицы неисправностей:

$$F(T) = (\bigvee_i F_i) \oplus A = 0.$$

Здесь решением является такое сочетание столбцов, участвующих в векторной операции логического сложения, которое в совокупности дает результат, равный вектору экспериментальной проверки. Поскольку операция времязатратная, то для нее следует использовать мультипроцессор Хассе, ориентированный на взятие булеана в почти параллельном режиме.

Подводя итог, следует представить модель процесса диагностирования цифровых устройств, которая содержит функциональные преобразователи, связанные с выполнением следующих шагов (рис. 6.2.2):

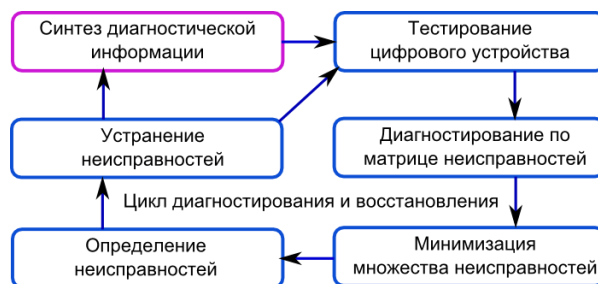


Рис. 6.2.2. Цикл диагностирования и ремонта логических блоков

1. Препроцессирование сводится к генерированию исходной диагностической информации в виде теста диагностирования, таблицы неисправностей и матрицы достижимостей цифровой системы.
2. Тестирование реального устройства на основе использования промышленного симулятора в целях сравнения фактических реакций устройства с эталонными значениями по наблюдаемым линиям-ассерциям, что дает возможность сформировать матрицу выходных реакций или вектор экспериментальной проверки в двоичном алфавите.
3. Вычисление матрицы активности графовой структуры на каждом входном тестовом наборе, равной по размерности таблице неисправностей, с помо-

стью матрицы экспериментальной проверки и матрицы достижимостей, что дает возможность существенно сократить область подозреваемых дефектов.

4. Модификация содержимого таблицы неисправностей путем ее маскирования матрицей активности графовой структуры, в целях определения только тех неисправностей, которые действительно формируют матрицу экспериментальной проверки в процессе диагностирования.

5. Выполнение процедуры логического умножения над строками таблицы неисправностей для получения вектора подозреваемых дефектов.

6. Восстановление работоспособности цифрового устройства путем переадресации неисправных логических компонентов на их аналоги из ремонтного запаса и повторение процесса тестового диагностирования.

Таким образом, новизна предложенного метода диагностирования дефектов заключается в использовании для получения диагноза единственной параллельной операции логического умножения, что в сочетании со структурным маскированием неисправностей дает преимущества перед аналогами в части увеличения быстродействия и повышения глубины диагностирования.

6.2.3. Кубитное моделирование цифровых систем. Рассматриваются структуры данных, эффективные с точки зрения программной или аппаратной реализации исправного интерпретативного моделирования дискретных систем, описанных в форме кубитных векторов состояний выходов примитивов. Для описания цифровой схемы, представленной на рис. 6.2.3, традиционно используется структура взаимосвязанных элементов и кубические покрытия (таблицы истинности) логического элементов.

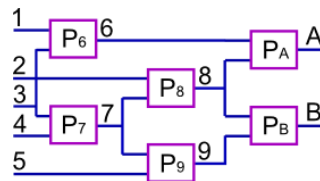


Рис. 6.2.3. Фрагмент цифровой схемы

Цель предлагаемого метода кубитного моделирования – заменить таблицы истинности компонентов цифрового устройства векторами состояний выходов. Пусть функциональный примитив с номером P_6 имеет следующую таблицу истинности:

$$P_6 = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ \hline \end{array}$$

Данное покрытие логического элемента можно трансформировать путем унитарного кодирования входных векторов на основе использования двухтактного алфавита [29-32, 36]. Символы и их коды, предназначенные для описания автоматных переменных, представляют собой булеан на универ-

суме из четырех примитивов, что соответствует формату вектора, содержащего два кубита:

$V^*(Y) = \{Q=(1000), E=(0100), H=(0010), J=(0001), O=\{Q,H\}=(1010),$
 $I=\{E,J\}=(0101), A=\{Q,E\}=(1100), B=\{H,J\}=(0011), S=\{Q,J\}=(1001),$
 $P=\{E,H\}=(0110), C=\{E,H,J\}=(1110), F=\{Q,H,J\}=(1011), L=\{Q,E,J\}=(1101),$
 $V=\{Q,E,H\}=(1110), Y=\{Q,E,H,J\}=(1111), U=(0000)\}.$

С помощью двухтактного алфавита любое покрытие функционального примитива путем кодирования входных наборов и последующего объединения символов всегда можно представить двумя кубами или даже одним, учитывая, что кубы взаимно инверсны:

$$P_6 = \begin{bmatrix} 00 & 1 \\ 01 & 1 \\ 10 & 1 \\ 11 & 0 \end{bmatrix} = \begin{bmatrix} Q & 1 \\ E & 1 \\ H & 1 \\ J & 0 \end{bmatrix} = \begin{bmatrix} V & 1 \\ J & 0 \end{bmatrix} = \begin{bmatrix} 1110 & 1 \\ 0001 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$$

Два куба показывают не только все решения, но и инверсию сигналов на выходе, что интересно с позиции активизации всех логических путей в схемной структуре при синтезе тестов. Например, для изменения состояния выхода необходимо создать на входах пару следующих друг за другом условий, где в первом такте должны быть первые три вектора (адреса), а во втором – четвертый вектор, формируемый двумя входными переменными.

Для моделирования исправного поведения достаточно иметь только один куб (нулевой или единичный), поскольку второй всегда является дополнением к первому. Следовательно, ориентируясь, например, на единичный куб, формирующий на выходе 1, можно убрать бит состояния выхода примитива, что уменьшит размерность куба или модели примитива до количества адресуемых состояний элемента, где адрес есть вектор, составленный из двоичных значений входных переменных, по которому определяется состояние выхода примитива.

Кубитное покрытие или Q-покрытие есть векторная интерпретативная форма задания функциональности, где значение координаты определяет состояние выхода функции, соответствующее двоичному входному слову, формирующему адрес ячейки. Q-покрытие одновыходового примитива всегда представлено двумя взаимно инверсными кубами (векторами), размерность которых равна степени двойки от числа входных переменных, где единичное значение координаты определяет участие адреса рассматриваемого бита в формировании соответствующего (0,1) состояния выхода примитива. Кубитные модели примитивов требуют создания новой теории моделирования, прямой и обратной импликации, синтеза тестов, моделирования неисправностей, поиска дефектов. Здесь и далее представлены основные процедуры исправного моделирования на основе манипулирования адресами, неявно представленными в координатах кубов Q-покрытия.

Модель для анализа цифровой системы на основе использования кубитных структур данных может быть описана четырьмя компонентами:

$$\begin{aligned}
F &= \langle L, M, X, Q \rangle, \\
L &= (L_1, L_2, \dots, L_j, \dots, L_n); \\
M &= (M_1, M_2, \dots, M_j, \dots, M_n); \\
X &= (X_{n_x+1}, X_{n_x+2}, \dots, X_{n_x+i}, \dots, X_n); \\
Q &= (Q_{n_x+1}, Q_{n_x+2}, \dots, Q_{n_x+i}, \dots, Q_n).
\end{aligned}$$

Здесь представлены соответственно: L – вектор идентификаторов эквивалентных линий схемы цифровой системы, который ввиду своей тривиальности может быть исключен из модели, но при этом необходимо иметь число входных переменных устройства и общее количество линий; M – вектор моделирования состояний всех линий схемы; X – упорядоченная совокупность векторов входных переменных каждого примитива схемы, привязанных к номерам выходов, Q – совокупность Q -покрытий примитивов, строго привязанных к номерам выходов и входным переменным примитивов; n – число линий в схеме, n_x – количество входных переменных.

В качестве примера кубитного задания модели цифрового устройства $F = \langle L, M, X, Q \rangle$, представленного на рис. 3, ниже приведен вариант структурной таблицы описания схемы для анализа исправного поведения:

L	1	2	3	4	5	6	7	8	9	A	B
M	1	1	1	1	1	0	1	0	1	1	0
X	.	.	.	.	.	13	34	27	75	68	89
Q	.	.	.	.	.	1	0	1	1	1	1
	.	.	.	.	.	1	1	0	0	0	0
	.	.	.	.	.	1	1	0	0	1	1
	.	.	.	.	.	0	1	0	1	0	1

Метод кубитного моделирования исправного поведения сводится к определению значения выхода элемента по адресу, формируемому конкатенацией двоичных состояний входных переменных каждого примитива цифровой системы

$M(Y_i) = Q_i[M(X_{i1} * X_{i2} \dots * X_{ij} \dots * X_{ik_i})]$. Здесь k_i – число входных линий в примитиве с номером i . Поскольку номера невходных линий вектора L однозначно идентифицируют по выходам обрабатываемые примитивы, то формула моделирования может быть приведена к циклу определения состояний всех невходных переменных:

$$M_i = Q_i[M(X_{i1} * X_{i2} \dots * X_{ij} \dots * X_{ik_i})] = Q_i[M(A_i)], \quad i = \overline{n_x + 1, n}.$$

Здесь процесс моделирования связан с конкатенированным формированием адреса бита в кубите функциональности, который определяет состояние примитива или невходной линии цифровой структуры, начиная с номера $i = n_x + 1$. Если переменные создают не двоичный адрес, то в данном случае существует возможность формирования состояния выхода логического эле-

мента в троичном алфавите символом X. Состояния выходов формируются идеально примитивной процедурой обработки кубита примитива

$$M_i = Q_i[M(X_i)]$$

на основе простых итераций или итераций Зейделя [27, 31]. Во втором случае необходима препроцессорная процедура ранжирования линий и примитивов схемы, которая позволяет существенно уменьшить количество проходов по элементам схемы для достижения сходимости, когда фиксируется равенство состояний всех линий в двух соседних итерациях. Кроме того, ранжирование примитивов по уровням формирования выходов дает возможность существенно повысить быстродействие моделирования за счет параллельной обработки функциональных элементов одного уровня. Например, для схемы, представленной на рис. 6, одновременно можно обрабатывать элементы с номерами 6,7, затем – 8,9 и далее – А,В. В первом случае, когда используются простые итерации, ранжирования не требуется, но платой за простоту алгоритма моделирования является существенно большее число итеративных проходов по примитивам схемы для достижения упомянутого критерия сходимости. Вычислительная сложность предложенного Q-метода моделирования на основе кубитных функциональностей определяется процедурами формирования адреса – входного вектора, содержащего k_i переменных, для каждого i-го примитива $[(r+w) \times k_i]$, считыванием бита из кубит-вектора по конкатенированному адресу и записью $(r+w)$ данного бита в вектор моделирования:

$$\eta = \sum_{i=n_x+1}^n \{[(r+w) \times k_i] + (r+w)\} = \sum_{i=n_x+1}^n [(r+w) \times (k_i + 1)] = (r+w) \times \sum_{i=n_x+1}^n (k_i + 1).$$

Время моделирования одного тест-вектора Q-методом, при условии, что цифровая схема, составленная из 900 четырехвыходовых примитивов, имеет параметры: $r = w = 5ns$, $k_i = 4$, $n_x = 100$, $n = 1000$, равно 45 микросекунд:

$$\eta = (r+w) \times \sum_{i=n_x+1}^n (k_i + 1) = (5+5) \times 900 \times (4+1) = 10 \times 900 \times 5 = 45000ns = 45\mu s.$$

Это означает, что быстродействие интерпретативного Q-метода моделирования дает возможность для данной схемы обработать за одну секунду 22 222 входных наборов. При этом цифровое устройство имеет существенное преимущество – сервисную функцию online восстановления работоспособности в случае отказа примитива путем его переадресации на запасной элемент.

Для синтеза квазиоптимальных структур данных комбинационного устройства необходимо использовать следующие правила:

1) Ранжированная схема цифрового устройства по структурной глубине для моделирования по способу Зейделя должна иметь по возможности однотипные примитивы в каждом уровне (слое) срабатывания.

2) В каждом уровне желательно иметь одинаковое число примитивов. Поэтому синтез цифрового устройства следует ориентировать на создание прямоугольной (матричной) структуры однотипных логических элементов.

3) Реализация комбинационных примитивов предполагает использование адресуемых элементов памяти, имеющих место быть в программируемых логических устройствах (FPGA, CPLD), широко используемых для создания прототипов.

4) Формирование для каждого уровня комбинационного устройства ремонтных примитивов для восстановления работоспособности в режиме online из расчета – один запасной элемент на каждый тип компонента, используемый в уровне.

5) Стоимость аппаратных затрат для реализации комбинационного устройства, ориентированного на высокое быстродействие, должна определяться суммой всех примитивов, привязанных к уровням комбинационного устройства, дополненной линейкой запасных элементов по одному для каждого слоя (при условии существования в каждом слое одинаковых примитивов):

$$Q = \sum_{i=1, n}^{j=1, m} P_{ij} + n.$$

6) Реализация комбинационного устройства, ориентированного на минимизацию аппаратных затрат, определяется суммой всех типов примитивов, инвариантных к уровням комбинационного устройства, дополненной линейкой запасных элементов по одному для каждого типа:

$$Q = \sum_{i=1}^m P_i + m.$$

7) Обработка матрицы комбинационных элементов с помощью процессорной линейки примитивов, число которых равно мощности максимального уровня или слоя в прямоугольной структуре, что обеспечивает условия для параллельной обработки всех примитивов в каждом уровне элементов в целях повышения быстродействия комбинационного прототипа, реализуемого в PLD.

Таким образом, новизна предложенного Q-метода интерпретативного исправного моделирования цифровых схем заключается в существенном повышении быстродействия и уменьшении объемов структур данных за счет замены таблиц истинности на Q-покрытия, что практически делает разработку конкурентоспособной с технологиями компилятивного моделирования.

6.2.4. Восстановление работоспособности комбинационных устройств. Немногочисленные работы, посвященные восстановлению работоспособности логических схем [28, 37, 38], описывают две идеи. Первая заключается в реконфигурации структуры логических элементов в режиме offline, которая

обеспечивает возможность замены каждого из неисправных примитивов. Вторая создает условия для замены неисправных элементов путем использования запасных логических компонентов и мультиплексоров для переадресации отказавших примитивов.

Структуры кубитных данных модифицируются в сторону дополнения строкой типов примитивов $F = \langle L, M, X, P, Q \rangle$, $P = (P_1, P_2, \dots, P_1, \dots, P_m)$, задействованных при синтезе цифровой системы, если необходимо в процессе функционирования выполнять ремонт или восстановление работоспособности за счет введения запасных примитивов, которые, так же как и основные, реализуются на основе элементов памяти. На рис. 6.2.4 изображен пример схемной структуры из адресуемых и трех запасных элементов. Структуры данных, соответствующие данной схеме с тремя дополнительными элементами представлены здесь же:

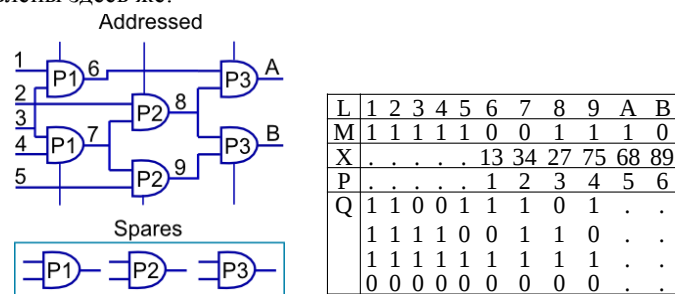


Рис. 6.2.4. Пример схемной структуры из адресуемых и запасных элементов

Таблица (см. рис. 6.2.4) оперирует номерами структурных примитивов, что дает возможность заменить любой отказавший элемент исправным из ремонтного запаса путем изменения адресного номера в строке примитивов P. Ремонтные элементы в данной таблице начинаются со столбца номер 7.

В следующей таблице представлены строка типов логических элементов, а также адреса типов этих примитивов, отмеченные номерами:

L	1	2	3	4	5	6	7	8	9	A	B
M	1	1	1	1	1	0	0	1	1	1	0
X	.	.	.	.	.	13	34	27	75	68	89
P	.	.	.	.	.	1	1	2	2	3	3
Q	1	0	1	1	0	1	.	.	.	.	.
	1	1	0	1	1	0	.	.	.	.	.
	1	1	1	1	1	1	.	.	.	.	.
	0	0	0	0	0	0	.	.	.	.	.

Данная структура данных ориентирована на программную реализацию моделирования, а ремонтные примитивы начинаются с номера 4. Если существует возможность перепрограммирования логики в элементе памяти с одинаковым числом входных переменных, то данную процедуру следует выполнять после фиксации неисправного элемента, когда становится известно

– какой элемент в структуре и какой тип примитива отказал. Процедура восстановления работоспособности ориентирована на PLD-реализацию цифровых систем. Если кубитные модели схем не имеют запасных примитивов, то соответствующий формат таблиц будет иметь следующий вид:

L	1	2	3	4	5	6	7	8	9	A	B
M	1	1	1	1	1	0	0	1	1	1	0
X	.	.	.	.	.	13	34	27	75	68	89
P	.	.	.	.	.	1	1	2	2	3	3
Q	1	0	1	.	.	.	.	.	.	.	.
	1	1	0	.	.	.	.	.	.	.	.
	1	1	1	.	.	.	.	.	.	.	.
	0	0	0	.	.	.	.	.	.	.	.

Таким образом, кубитные структуры данных ориентированы на компактность описания функционалов цифрового изделия векторами состояний выходов, повышение быстродействия процедур моделирования за счет адресации состояний выходов примитивов, а также на восстановление работоспособности отдельных логических элементов, благодаря их реализации в элементах памяти PLD или в форме программных модулей. Очень важно, что в последнем случае не нужно хранить ремонтные примитивы, поскольку предложенные здесь интерпретативные структуры табличных данных изначально ориентированы на технологические удобства устранения дефектов в процессе функционирования прототипа цифрового изделия.

Обработка схемы в кристалле сводится к определению адреса, составленного двоичными битами вектора моделирования, по которому находится значение логической функции. Каждый примитив имеет цикл обработки, содержащий три процедуры:

- 1) Адресное считывание номеров входных переменных из соответствующего столбца матрицы X для формирования адреса состояния входной переменной вектора моделирования: $A = X_{ij}$, $i = \overline{1, n}$; $j = \overline{1, s_p - 1}$;
- 2) Формирование адреса (двоичного кода) для вычисления логической функции путем конкатенации соответствующих состояний входных переменных в векторе моделирования $A = M(X_{ij}) * M(X_{ir})$;
- 3) Запись результата выполнения логической функции как состояния выхода в соответствующий разряд вектора моделирования $M(X_{is_p}) = P[M(X_{ij}) * M(X_{ir})]$.

Процесс обработки всех примитивов схемы в данном случае является строго последовательным, что представляет собой существенное замедление процедуры формирования состояний выходных переменных. Однако уменьшение быстродействия можно считать платой за сервис встроенного и автономного восстановления работоспособности цифровой структуры, который является одним из этапов функционирования инфраструктуры обслуживания SoC, представленной на рис. 6.2.5.

определения состояния его выхода с последующей записью в соответствующую координату вектора моделирования: $M(X_{k+1})$:

$$M(X_{k+1}) = \{P_i, Q_i\} \left[\#_{j=1}^k M(X_{ij}) \right].$$

5. Повторение пунктов 3 и 4 в целях получения состояний выходов всех логических элементов до выполнения условия: $i = n$.

6. Повторение пунктов 2–4 в целях моделирования всех входных тестовых (рабочих) наборов, до выполнения равенства: $t = \eta$, где η – длина теста.

7. Окончание процесса моделирования цифрового устройства.

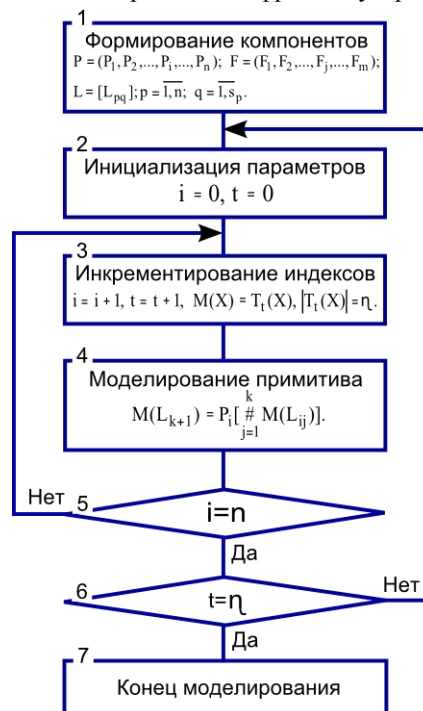


Рис. 6.2.6. Граф-схема алгоритма управления процессом моделирования

Таким образом, научная новизна предложенной модели цифровой системы заключается во введении в структуру устройства избыточных ремонтных компонентов и управляющего автомата, ориентированного на последовательную обработку комбинационных примитивов, что дает возможность осуществлять процедуру переадресации примитивов в случае отказа одного из них. Нетрудно создать аналогичные автоматы для параллельной обработки слоев из примитивов ранжированной схемы, что максимально приблизит быстроедействие устройства к его реализации в кристаллах PLD.

6.2.5. Выводы

В данном разделе описаны основы применения кубитных структур, ориентированных на параллельное вычисление теоретико-множественных по сути данных, которые формируются в следующие несколько пунктов:

1. Усовершенствован метод диагностирования дефектов цифровых систем за счет использования единственной параллельной операции логического умножения, что в сочетании со структурным маскированием неисправностей дает преимущества перед аналогами в части компактности представления данных, увеличения быстродействия и повышения глубины диагностирования.
2. Предложен новый Q-метод интерпретативного исправного моделирования цифровых схем, который характеризуется использованием компактных Q-покрытий вместо таблиц истинности, что дает возможность существенно повысить быстродействие анализа за счет адресного формирования выходов функциональных примитивов и уменьшить объемы структур данных, что практически делает метод конкурентоспособным с технологиями компилятивного моделирования.
3. Усовершенствована модель цифровой системы путем дополнения в структуру устройства избыточных ремонтных компонентов и управляющего автомата, ориентированного на последовательную обработку комбинационных примитивов, что дает возможность осуществлять процедуру переадресации отказавших примитивов в режиме штатного функционирования.
4. Показаны примеры использования кубитных структур данных и процедур формирования адресов для моделирования цифровых схем и решения задач диагностирования путем использования векторных параллельных логических операций и ремонта неисправных модулей на основе адресуемых логических примитивов.
5. Основная инновационная идея квантовых или кубитных вычислений заключается в переходе от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение (точку), к квантовым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений.

ЗАКЛЮЧЕНИЕ

В работе представлены математические основы теории надежности микро-систем. Дан аналитический обзор современного состояния рынка микро-электромеханических систем (МЭМС). Используются лучшие зарубежные и отечественные издания, государственные стандарты, а также работы ведущих ученых и специалистов в области теории надежности. Предлагаются кубитные модели и методы повышения тестопригодности программных и аппаратных средств анализа цифровых устройств за счет увеличения размерности структур данных и памяти. Введены основные понятия, термины и определения, необходимые для имплементации квантовых вычислений в практику анализа виртуальных компьютеров. Представлены результаты исследований, касающиеся моделей и методов диагностирования цифровых систем, моделирования исправного поведения, восстановления работоспособности отказавших примитивов.

Описана инновационная идея квантовых вычислений, которая характеризуется переходом от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение (точку) к логическим регистровым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений, что дает возможность определить новые перспективы на пути создания высокопроизводительных компьютеров параллельного анализа и синтеза структур и сервисов дискретного киберпространства.

Предложены кубитные модели описания цифровых систем и компонентов, которые характеризуются компактностью описания таблиц истинности в форме Q-покрытий благодаря унитарному кодированию входных состояний, что дает возможность повысить быстродействие программных и аппаратных средств интерпретативного моделирования вычислительных устройств за счет адресной реализации анализа логических примитивов.

Представлена матричная модель кубитных примитивов для реализации комбинационных схем, которая характеризуется адресным объединением Q-покрытий на элементах памяти, «мягко» соединенных в цифровую схему с помощью вектора состояний линий, что дает возможность восстанавливать работоспособность отказавших логических примитивов в реальном времени с помощью их переадресации на запасные компоненты при достаточно высоком быстродействии функционирования вычислительного устройства.

Введена гибкая автоматная MQT-модель компьютера, которая характеризуется использованием только адресуемых структур памяти и операции тран-

зации для программной и аппаратной реализации комбинационных и последовательностных функциональностей, что дает возможность создавать быстродействующие и надежные вычислители для создания сервисов киберпространства на основе параллельных логических операций и ремонта неисправных адресуемых функциональных примитивов.

Усовершенствован метод диагностирования дефектов цифровых систем за счет использования единственной параллельной операции логического умножения, что в сочетании со структурным маскированием неисправностей дает преимущества перед аналогами в части компактности представления данных, увеличения быстродействия и повышения глубины диагностирования.

Предложен новый Q-метод интерпретативного исправного моделирования цифровых схем, который характеризуется использованием компактных Q-покрытий вместо таблиц истинности, что дает возможность существенно повысить быстродействие анализа за счет адресного формирования выходов функциональных примитивов и уменьшить объемы структур данных, что практически делает метод конкурентоспособным с технологиями компилятивного моделирования.

Усовершенствована модель цифровой системы путем дополнения в структуру устройства избыточных ремонтных компонентов и управляющего автомата, ориентированного на последовательную обработку комбинационных примитивов, что дает возможность осуществлять процедуру переадресации

Показаны примеры использования кубитных структур данных и процедур формирования адресов для моделирования цифровых схем и решения задач диагностирования путем использования векторных параллельных логических операций и ремонта неисправных модулей на основе адресуемых логических примитивов.

Представлена концепция системного проектирования программных и аппаратных облачных сервисов киберпространства на основе MQT-автомата, которая характеризуется применением транзакций на кубитных адресно связанных компонентах вычислителя с помощью вектора состояний (моделирования), что дает возможность программировать логические функциональности в кубитных примитивах интерпретативной структуры цифрового устройства.

Рыночная привлекательность MQT-компьютера определяется: примитивизмом его реализации в программном и аппаратном исполнении; высоким уровнем использования памяти в структуре современных вычислителей,

достигающим 94% от площади кристалла; и как следствие – уменьшением влияния комбинационной логики на быстродействие системы в целом; повышением надежности компьютеров за счет онлайн-ремонта адресуемых элементов памяти, включая логические примитивы. Существующее неприятие MQT-компьютера рынком связано с лоббированием вычислителей на основе жесткой комбинационной логики в VLSI-проектах со стороны компаний, разрабатывающих процессоры на основе технологий reusable logic.

Направления будущих исследований: Синтез кубитных моделей цифровых систем на основе использования Q-покрытий, а также разложение (анализ) функциональностей на кубитные компоненты. Моделирование неисправностей и синтез тестов на основе использования Q-покрытий компонентов цифровых систем. Разработка специализированных программных приложений для решения задач синтеза комбинационных устройств на основе использования матричной модели MQT-процессора, ориентированного на высокий параллелизм решения практических задач и восстановление работоспособности устройства в реальном времени.

8. СПИСОК ЛИТЕРАТУРЫ

1. В. Майская. МЭМС РЫНОК. Стойкий продолжительный рост // ЭЛЕКТРОНИКА. №5. 2012. <http://www.electronics.ru/journal/article/3330>
2. <https://technology.ihs.com/532252/mems-market-tracker-q1-2015>
3. Jeff Dorsch. The Internet of Things Needs MEMS and Sensors – Plus More Security and Better Software. <http://electronics360.globalspec.com/article/6039/the-internet-of-things-needs-mems-and-sensors-plus-more-security-and-better-software>
4. Hartzell A. L., Da Silva M. G., Shea H. MEMS Reliability. –Springer. –2011.– 289 p.
5. Matej Andrejašič. MEMS ACCELEROMETERS // University of Ljubljana, Faculty for mathematics and physics, Department of physics. 2008.–17 p.
6. Староверов К. Новое семейство одно- и двухосевых гироскопов // Новости электроники. 2009. №14. Электронный ресурс <http://www.compel.ru/lib/ne/2009/14/5-novoe-semeystvo-odno-i-dvuhosevyih-giroskopov>
7. W. Merlijn van Spengen. MEMS reliability from a failure mechanisms perspective // Microelectronics Reliability 43 (2003) 1049–1060 // www.elsevier.com/locate/microrel
8. Sandeep Akkaraju. N/MEMS Design Methodologies. Efficient workflow for robust, first-timeright design. – White Paper. – IntelliSense Woburn, MA.– 22 p.
9. Fan, X. Fault diagnosis of VLSI designs: cell internal faults and volume diagnosis throughput / Xiaoxin Fan // PhD (Doctor of Philosophy) thesis, University of Iowa.– 2012.– 134 p.
10. Stanisavljevi, M. Reliability of Nanoscale Circuits and Systems / M. Stanisavljevi, M. Schmid, Y. Leblebici.– Springer.– 2011.– 240 p.
11. Pomeranz, I. Transition Path Delay Faults: A New Path Delay Fault Model for Small and Large Delay Defects / I. Pomeranz, S.M. Reddy // IEEE Transactions on Very Large Scale Integration (VLSI) Systems.– 2008.– Vol.16, No.1.– P. 98–107.
12. Harris, I.G. Hardware-Software Covalidation: Fault Models and Test Generation / Ian G. Harris // Design and Test of Computers.– Vol. 20, Num. 4.– July-August 2003.– 12 p. Электронный ресурс: <http://www.npk-photonica.ru/content/products/mems>
13. ДСТУ 2860-94 Надійність техніки. Терміни та визначення.
14. Расулова С.С. НАДЕЖНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ. Конспект лекций. Ташкент. 2007. 116 с.
15. Ермаков А.А. Основы надежности информационных систем: учебное пособие. – Иркутск: ИрГУПС, 2006.– 151с.

16. System Analysis Reference. Reliability, Availability & Optimization. ReliaSoft Corporation Worldwide Headquarters. 1450 South Eastside Loop. Tucson, Arizona 85710-6703, USA. 281 p.
17. ДСТУ 2861-94 Надійність техніки. Аналіз надійності. Основні положення.
18. FTA. Дерево отказов, как метод структурного анализа. Электронный ресурс <http://www.itexpert.ru/rus/ITEMS/77-30/>
19. Анализ дерева неисправностей FTA. Электронный ресурс <http://metrology.com.ua/risk-menedzhment/opisanie-metodov-otsenki-riskov/analiz-dereva-neispravnostej-fta>
20. ОСИПОВА Н.А. Техногенные системы и экологический риск: Учебное пособие. Ч.1. Томск: Изд. ТПУ, 2005. – 112 с.
21. Левин Б. Р. Теория надежности радиотехнических систем (математические основы). Учебное пособие для вузов. М., «Сов. Радио» 1978. 264 с.
22. Проект національного стандарту України. Надійність техніки. Оцінювання та прогнозування надійності за результатами випробувань і (або) експлуатації в умовах малої статистики відмов. – Київ: ДП «УкрНДНЦ». – 2015. – 58 с.
23. Michael A. Nielsen & Isaac L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press. 2010. 676p.
24. Stig Stenholm, Kalle-Antti Suominen. Quantum approach to informatics. John Wiley & Sons, Inc., 2005. 249p.
25. Mark G. Whitney. Practical Fault Tolerance for Quantum Circuits. PhD dissertation. University of California, Berkeley. 2009. 229p.
26. Mikio Nishihara. Quantum Computing. An Overview. Higashi-Osaka. Kinki University, 2010. 53p.
27. Горбатов В.А. Основы дискретной математики. – М. : Высшая школа. – 1986. – 311 с.
28. Hahanov V.I., Litvinova E.I., Chumachenko S.V., Baghdadi Ammar Awni Abbas, Eshetie Abebech, Mandefro. Qubit Model for solving the coverage problem // Proc. of IEEE East-West Design and Test Symposium. Kharkov. 14-17 September 2012. P.142 - 144.
29. Хаханов В.И., Мурад Али Аббас, Литвинова Е.И., Гузь О.А., Хаханова И.В. Квантовые модели вычислительных процессов // Радиоэлектроника и информатика. 2011. №3. С.35-40.
30. Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. – Харьков: ХНУРЕ. – 2009. – 484 с.
31. Хаханов В.И. Техническая диагностика цифровых и микропроцессорных структур. – К.: ИСХО. – 1995. – 242 с.
32. Бондаренко М.Ф., Хаханов В.И., Литвинова Е.И. Структура логического ассоциативного мультипроцессора. Автоматика и телемеханика. 2012. № 10. С. 71-92.

33. Борисовец Б. Э. , Шаршунов С. Г. Общая модель и синтез тестов для механизмов управления межрегистровым обменом данными в микропроцессорах // Автоматика и телемеханика, 1992, № 8, 142–149.
34. Koal T., Scheit D., Vierhaus H.T. A comprehensive scheme for logic self repair // Conf. Proc. on Signal
35. Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S. Information analysis infrastructure for diagnosis // Information an int. interdisciplinary journal. 2011. Japan. Vol.14. № 7. P. 2419-2433.
36. Vladimir Hahanov, Alexander Barkalov and Marian Adamsky. Infrastructure intellectual property for SoC simulation and diagnosis service. Springer, 2011. P. 289-330.
37. Хаханов В.И., Литвинова Е.И., Хаханова И.В., Murad Ali Abbas. Инфраструктура встроенного восстановления логических PLD-схем// Радиоэлектроника и информатика. 2012. №2. С. 54-57.
38. Hahanov V., Litvinova E., Gharibi W., Murad Ali Abbas. Qubit models for SoC Synthesis Parallel and cloud computing. USA. 2012. Vol.1. Iss 1. P. 16-20.