Министерство образования Республики Беларусь
**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА**

Кафедра микропроцессорной техники
и информационно-управляющих систем

К. А. БОЧКОВ, С. Н. ХАРЛАП

# МЕТОДЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ В МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ ЖЕЛЕЗНОДОРОЖНОЙ АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

*Допущено Министерством образования Республики Беларусь*
*в качестве учебного пособия для студентов*
*транспортных специальностей высших учебных заведений*

Гомель 2001

УДК 656.25
ББК 39.275
Б 866

Р е ц е н з е н т ы : заведующий кафедрой «Математические проблемы управления» ГГУ, докт. техн. наук, профессор **И.В. Максимей,** начальник службы сигнализации и связи **И.И. Аксютик**

**Бочков К.А., Харлап С.Н.**

Б 866 Методы обеспечения безопасности в микропроцессорных системах железнодорожной автоматики и телемеханики: Учеб. пособие для студентов транспортных специальностей высших учебных заведений – Гомель: БелГУТ, 2001. – 84 с.
ISBN 985-6550-60-2

Раскрыты основные принципы построения микропроцессорных информационно-управляющих систем железнодорожной автоматики и телемеханики. Особое внимание уделено обеспечению безопасности микроэлектронных и микропроцессорных систем.

Предназначено для студентов транспортных специальностей высших учебных заведений.

**УДК 656.25**
**ББК 39.275**



ISBN 985-6550-60-2

## ВВЕДЕНИЕ

В современных условиях наиболее распространенными в эксплуатации системами железнодорожной автоматики и телемеханики (СЖАТ) остаются релейные системы, которые удовлетворяют большинству технических и функциональных требований, предъявляемых к таким системам. В 80-е годы были разработаны новые релейные системы, принятые к типовому проектированию (электрическая централизация с индустриальной системой монтажа и др.).

Однако релейной технике присущи недостатки, которые ограничивают ее применение в современных СЖАТ. Область этих ограничений расширяется и будет расширяться со временем. К недостаткам относятся: невысокое быстродействие, большие размеры, большая материалоемкость и значительный расход дефицитных материалов, невысокая надежность. Поэтому интегрировать их в комплекс устройств управления движением, который во все большей мере основывается на системах передачи информации, можно только ценой очень высоких затрат.

Решение данной проблемы возможно при использовании микроэлектронной, микропроцессорной и компьютерной техники для построения СЖАТ. С начала 80-х годов разрабатываются и внедряются на железных дорогах мира новые системы микропроцессорных и компьютерных централизаций стрелок и сигналов, диспетчерских централизаций, микроэлектронные и микропроцессорные системы интервального регулирования движения поездов и другие СЖАТ. По сравнению с релейными системами современные микропроцессорные СЖАТ имеют следующие преимущества: резервирование технических средств, обеспечивающее надежное функционирование системы при выходе из строя отдельных компонентов; построение системы из отдельных независимых модулей с простой структурой; значительно меньшие масса и габариты модулей системы, что позволяет резко снизить затраты на их размещение; более высокая эксплуатационная готовность системы; более простое техническое обслуживание и в связи с этим меньшие эксплуатационные расходы [1]. Поэтому, именно микроэлектронная, микропроцессорная и компьютерная техника определяет на ближайшее будущее развитие средств железнодорожной автоматики и телемеханики.

Целесообразность и необходимость использования микроэлектронной, микропроцессорной и компьютерной элементной базы связаны со следующими основными тенденциями развития СЖАТ [1]:

1) существенное повышение требований к функциям разрабатываемых систем. Обязательно наличие развитого диагностического обеспечения, обширных банков данных, удобных автоматизированных рабочих мест, автоматических устройств регистрации действий операторов и др. Новые задачи сложнее, и их удобнее решать с использованием компьютерной техники;

2) структурное усложнение СЖАТ. Из-за увеличения числа решаемых функций значительно (в десятки и более раз) увеличивается число элементов и блоков, из которых состоит система. Поэтому, новые СЖАТ требуется разрабатывать с использованием БИС, СБИС и ПЭВМ;

3) повышение требований к безотказности и отказоустойчивости систем. Так как все больше функций передается от человека устройствам автоматики, то возрастает экономический ущерб в результате отказа СЖАТ. Основной путь повышения надежности – введение избыточности. Избыточность современных надежных систем состоит в удвоении и даже утроении объема аппаратного или программного обеспечения. При этом стоимость систем многократно возрастает, что делает релейную элементную базу экономически невыгодной;

4) повышение требований к контролепригодности СЖАТ. Применение БИС, СБИС и усложнение систем приводит к значительным трудностям при их обслуживании и ремонте. Новые системы должны иметь встроенные средства диагностики и контроля, позволяющие обнаруживать и локализировать отказавшие блоки. Ремонт таких систем заключается в замене отказавшего блока новым. Актуальной становится задача создания необслуживаемых СЖАТ, т. е. высоконадежных неремонтируемых систем, работающих до первого отказа.

Принципиальным отличием микропроцессорных СЖАТ является то, что в этих системах широко используются программные способы реализации алгоритмов управления и контроля, связанных с обеспечением безопасности движения поездов. Это требует принципиально новых подходов к обеспечению безопасности технических и программных средств СЖАТ.

# 1 ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

Выбор методов достижения требуемых показателей безотказности и безопасности осуществляется на основе анализа возможных отказов элементов СЖАТ в соответствии с принятой концепцией безопасности.

## 1.1 Концепция и стратегии обеспечения безопасности

Под *концепцией* безопасности понимается совокупность положений, в соответствии с которыми осуществляется построение безопасной системы. Такая концепция имеет фундаментальное значение, поскольку определяет основные принципы обеспечения безопасного функционирования СЖАТ. Концепция учитывает свойства элементной базы, средства контроля, структуру и алгоритм работы системы. На основе концепции безопасности устанавливаются критерии опасных отказов.

В релейных системах обеспечения безопасности отказы подразделяются на защитные и опасные. Поэтому в основе концепции безопасности релейных систем лежит принцип использования безопасного элемента. Например, в железнодорожной автоматике таким элементом является специальное железнодорожное реле, имеющее несимметричную характеристику отказов. Это значит, что наиболее вероятными являются искажения выходной информации типа $1 \rightarrow 0$, а искажения $0 \rightarrow 1$ маловероятны. Система строится в предположении, что отказы вида $0 \rightarrow 1$ отсутствуют, а остальные отказы должны переводить систему в защитное состояние (защитные отказы). Концепция безопасности для релейных систем формулируется следующим образом: «При построении ответственных цепей необходимо использовать только фронтовые контакты безопасного реле I класса надежности».

Появление сложных микроэлектронных и микропроцессорных элементов привело к выделению нового класса отказов – маскируемых. Дефекты технических средств, которые не приводят сразу к нарушению функционирования системы, называются маскируемыми и могут быть обнаруживаемыми и необнаруживаемыми. Необнаруживаемые отказы могут приводить к накоп-

лению неисправностей и, как следствие, к возможности появления опасных отказов.

При разработке систем на основе микропроцессорных БИС необходимо учитывать, что многие элементы используются многократно в ходе выполнения программы. Кроме того, отдельные отказы БИС (например, отказы в питающих выводах) могут приводить к искажениям в работе других БИС, к появлению новых паразитных связей между элементами. Последствия таких отказов могут проявиться в различных элементах микропроцессорных информационно-управляющих систем (МИУС), даже только косвенно связанных с источником отказа.

При разработке СЖАТ, кроме того, необходимо учитывать сбои в элементах памяти при воздействии электромагнитных помех. Такие сбои проявляются как константные неисправности.

Поэтому концепция безопасности для микроэлектронных систем такова: ”Одиночные дефекты аппаратных и программных средств не должны приводить к опасным отказам и должны обнаруживаться с заданной вероятностью при рабочих и тестовых воздействиях не позднее, чем в системе возникнет второй дефект.”

Для реализации концепции безопасности используют пять основных *стратегий*: безотказность, отказоустойчивость, безопасное поведение при отказах, безошибочность и помехоустойчивость.

*Безотказность СЖАТ* – свойство системы непрерывно сохранять работоспособное состояние в течение некоторого времени или наработки.

Стратегия безотказности подразумевает, что если в системе нет отказов, то она безопасна. Задачей стратегии безотказности является создание систем, у которых интенсивность отказов сравнима с нормируемой интенсивностью опасных отказов. В этом случае любой отказ можно считать опасным, никакой дополнительной защиты от отказов не требуется.

К основным показателям безотказности относятся: показатели невосстанавливаемых изделий ($P(t)$ – вероятность безотказной работы, $\lambda$ – интенсивность отказов, $T_o$ – среднее время наработки до отказа) и показатели восстанавливаемых изделий ($T_{\text{ср}}$ – средняя наработка на отказ, $K_{\text{г}}$ – коэффициент готовности).

Основные пути реализации стратегии безотказности:

1) минимизация логических схем;
2) снижение интенсивностей потока отказов элементов.

*Отказоустойчивость СЖАТ* – свойство системы сохранять работоспособность в случае отказа ее элементов благодаря резервным возможностям. Стратегия отказоустойчивости подразумевает, что если система правильно

выполняет свой алгоритм функционирования даже при наличии отказов, то она безопасна.

Отказоустойчивые системы нечувствительны к определенному числу отказов. Их еще называют *α-безотказными*. Это значит, что система работает правильно при наличии в ней $\alpha$ или менее отказов. Число $\alpha$ является показателем отказоустойчивости. Это качество системы резко повышает ее безотказность и безопасность, но требует введения большой избыточности в аппаратные и программные средства. Объем аппаратуры возрастает в таких системах в три и более раз. Поэтому отказоустойчивые релейные СЖАТ не разрабатывались. Для микроэлектронных СЖАТ отказоустойчивость становится одним из основных направлений развития.

Основные пути реализации стратегии отказоустойчивости:

1) резервирование;
2) диагностирование;
3) реконфигурация;
4) восстановление.

Отказоустойчивость базируется на резервировании. Остальные средства (диагностирование, восстановление и реконфигурация) только повышают ее эффективность.

В реальных МИУС, как правило, используют комплекс мер по обеспечению отказоустойчивости, причем в различных элементах и узлах системы можно использовать различные виды резервирования, отличающиеся степенью избыточности (кратностью резервирования). Используют различные методы и средства контроля, восстановления и реконфигурации.

Поэтому наиболее общим показателем отказоустойчивости является коэффициент отказоустойчивости

$$k_{\text{оу}} = T_{\text{нр}} / T_{\text{оу}},$$

где $T_{\text{нр}}$ – наработка между отказами нерезервированной системы;

$T_{\text{оу}}$ – наработка между отказами отказоустойчивой системы.

В качестве дополнительных показателей отказоустойчивости можно применять полноту резервирования элементов и узлов системы, полноту и достоверность контроля, вероятность восстановления резерва и т.п.

Первые две стратегии подразумевают, что система, которая правильно выполняет свой алгоритм функционирования, безопасна. Стратегия безопасного поведения при отказах используется специально для безопасных систем и заключается в переводе системы в защитное необратимое состояние при появлении отказа. Обратный переход в работоспособное состояние исключается и производится обычно с участием человека.

Основные пути реализации стратегии безопасного поведения при отказах:

1) использование самопроверяемых схем;
2) использование элементной базы с несимметричными характеристиками отказов.

При построении безопасных систем могут использоваться несколько различных стратегий одновременно (рисунок 1.1). Например, при построении микроэлектронных и микропроцессорных систем стратегия безопасного поведения применяется совместно со стратегией отказоустойчивости. Если при возникновении отказов система исчерпала резервные возможности и в результате деградации и реконфигурации перестала быть отказоустойчивой, то при появлении еще одного отказа она должна необратимо перейти в защитное состояние.

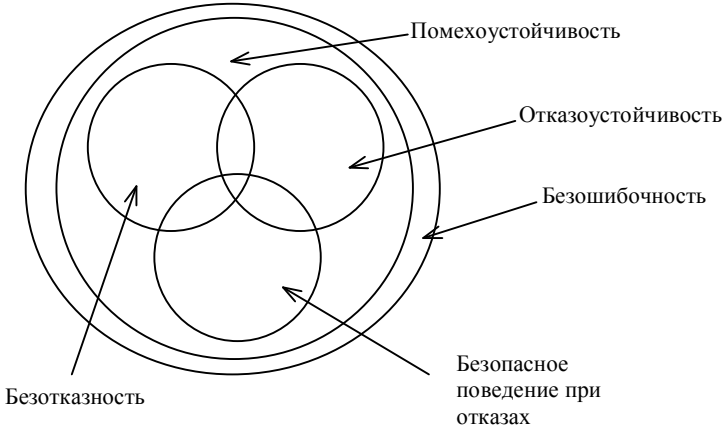



Рисунок 1.1 – Схема взаимодействия стратегий построения СЖАТ

Безопасность технических средств в разомкнутых системах управления, регулирования и контроля сильно зависит от человеческого фактора при разработке, изготовлении и эксплуатации системы. Поэтому для создания безопасных технических средств дополнительно используют стратегию безошибочности.

Стратегия безошибочности предполагает сведение к минимуму влияния на безопасность функционирования системы человеческого фактора (ошибок человека) при разработке, изготовлении и эксплуатации системы.

Основные пути реализации стратегии безошибочности:

1) при разработке и проектировании: поэтапное выполнение работ с верификацией результатов каждого этапа; документированность каждого этапа разработки; стандартизация и унификация; контролируемость процесса разработки; автоматизация проектирования;

2) организации оперативного управления: организация дружественного интерфейса пользователя; защита от несанкционированного доступа; преемственность по отношению к эксплуатируемым в настоящее время системам; наличие обратной связи от технических средств к пользователю; рациональное распределение функций между пользователем и техническими средствами для обеспечения умеренной загрузки оператора;

3) организации технического обслуживания и ремонта: обеспечение контролепригодности системы; простота представления контрольной и диагностической информации; интеллектуальная поддержка технического обслуживания и ремонта; дублирование информации о работоспособности системы.

Но даже разработанная по всем правилам построения безопасная СЖАТ может быть неработоспособна из-за ее низкой помехозащищенности. Поэтому на всех этапах разработки микропроцессорных и микроэлектронных систем необходимо проводить мероприятия по обеспечению заданного уровня помехоустойчивости.

*Помехоустойчивость* – это свойство аппаратуры, обеспечивающее защищенность ее от воздействия внешних электромагнитных влияний. В последнее время помехоустойчивость современных микроэлектронных систем обеспечения безопасности приобрела особенное значение в связи со следующими причинами:

1 По сравнению с релейными системами микроэлектронная элементная база в значительно большей степени подвержена воздействию электромагнитных помех. Это связано с увеличением сложности, уменьшением габаритных размеров микроэлектронных систем, повышением плотности монтажа, быстродействия и чувствительности элементной базы.

2 Системы управления ответственными технологическими процессами работают в сложной электромагнитной обстановке. Сбои в работе микроэлектронных систем происходят на порядок чаще чем отказы. В то же время последствия от влияния помех сравнимы с последствиями от отказов аппаратных средств и ошибок программного обеспечения. Поэтому при испытаниях на ЭМС микроэлектронных систем обеспечения безопасности актуальной является проблема анализа последствий сбоев и поиска сбоев, приводящих к нарушению безопасности функционирования систем (опасных сбоев).

3 Высокая сложность таких систем предполагает наличие большого числа состояний (тактов работы). Однако одно и то же электромагнитное воздействие в различных тактах функционирования системы может привести к различным последствиям. Поэтому требуется либо находить такты работы, наименее устойчивые к воздействию электромагнитных помех и испытывать систему, находящуюся в этих состояниях, либо повышать общий уровень помехоустойчивости.

Детерминированными показателями помехоустойчивости являются: восприимчивость элементной базы к электромагнитным помехам, которая выражается в статических (например, $U_{пор}$ – пороговое напряжение срабатывания) и динамических параметрах (например, $T_{пор}$ – пороговая длительность помех). Вероятностными показателями помехоустойчивости являются вероятность и интенсивность сбоев, наработка на сбой, наработка между сбоями.

Основными путями реализации стратегии помехоустойчивости являются:

- подавление электромагнитных помех в источнике возникновения;
- понижение восприимчивости к электромагнитным помехам аппаратуры МИУС;
- воздействие на паразитный канал проникновения помех.

Многообразие отказов и форм их проявления требует применения методов обеспечения безотказности и безопасности на различных функциональных уровнях СЖАТ.

### 1.2 Иерархия уровней обеспечения безопасности

Существуют два способа решения проблемы построения безопасных микропроцессорных информационно-управляющих систем:

1) создание и применение специализированных безопасных логических элементов, БИС, микропроцессоров и компьютеров, у которых вероятность возникновения определенного вида отказов настолько мала, что ею можно пренебречь;

2) использование коммерческих БИС, микропроцессоров и компьютеров, т. е. устройств не предназначенных специально для решения задач безопасности. В этом случае безопасность обеспечивается особенностями архитектуры системы, программным обеспечением и внешними контрольными схемами.

Первоначально в электронных системах автоматики использовался первый способ. При этом использовались электронные аналоги реле первого класса надежности. В настоящее время наиболее широко используется второй способ или комбинация обоих способов, когда функциональный блок выполняется с симметричными отказами, а контрольные схемы – с несимметричными.

Безопасность систем, построенных на элементах с симметричными отказами, обеспечивается введением различных видов избыточности. В настоящее время наиболее широко используются следующие виды избыточности: функциональная, структурная, временная и информационная.

Функциональная избыточность достигается за счет снижения эффективности функционирования и уровня автоматизации управления в случае отказа элементов системы при сохранении ее работоспособности. В этом случае

элементы, выполняющие функции, не связанные с обеспечением безопасности (различные сервисные функции), при отказе основных блоков системы берут на себя выполнение основных функций. Сервисные функции в этом случае не выполняются.

При структурной избыточности используется способ параллельной обработки информации в нескольких вычислительных каналах или с помощью нескольких программ в одном вычислительном канале. Структурная избыточность бывает аппаратной и программной.

Временная избыточность достигается выделением специальных промежутков времени во время работы системы для проведения контроля за правильностью функционирования системы в целом и ее составных частей. Временная избыточность может строиться на базе генераторов внешних тестовых воздействий и средств внешнего контроля или на базе средств самотестирования и самоконтроля.

Информационная избыточность достигается многократным повторением информации, критичной к вопросам безопасности, избыточным кодированием или многоканальной передачей информации.

Наиболее широко при построении систем микропроцессорных централизаций используются функциональная и структурная избыточность. Временная и информационная избыточность применяются только вместе с другими видами избыточности для повышения достоверности контроля правильности функционирования системы.

Мероприятия по защите от отказов в безопасных системах можно проводить на пяти уровнях защиты:

1) *аппаратный* уровень (самый низкий) реализуется, как правило, в устройствах управления объектом, контрольных схемах, схемах сравнения и мажоритарных схемах. Защита осуществляется с помощью элементов с несимметричными отказами, отказоустойчивых и самопроверяемых схем. Самопроверяемые схемы при возникновении в них неисправностей формируют на своих выходах сигнал ошибки;

2) на *информационном* уровне организуется защита от ошибок информации, хранящейся в памяти и передаваемой по каналам связи и шинам микроэлектронных систем. Основными методами защиты является применение памяти с аппаратным контролем паритета и использование корректирующих кодов при хранении и передаче информации. Наиболее часто используются коды с контролем на четность, равновесные, с повторением, с суммированием, арифметические, коды Хэмминга и др. [1];

3) *программном* уровне безопасность обеспечивается программными средствами, защищенными от отказов. Проблема повышения безотказности и безопасности программного обеспечения (ПО) более трудна и менее изуче-

на, чем для аппаратуры. Это связано со сложностью программных продуктов, большим разнообразием видов дефектов и их влиянием на процесс вычислений. Основными методами повышения надежности программ являются тестирование, верификация, создание самопроверяющихся программ;

4) уровне *архитектуры* необходимые показатели безопасности достигаются за счет использования структурного резервирования аппаратуры и ПО. В этом случае используется способ параллельной или последовательной обработки информации в нескольких вычислительных каналах, или с помощью нескольких программ в одном вычислительном канале;

5) уровне *интерфейса* (высшем) безопасность обеспечивается применением безопасного интерфейса с исполнительными объектами. Здесь разрабатываются специализированные безопасные устройства сопряжения с объектом управления (УСО).

На каждом из этих уровней можно осуществлять мероприятия по защите от отказов, позволяющие компенсировать последствия отказов, возникающих на более низких уровнях.

Далее рассматриваются методы и принципы обеспечения безопасности микроэлектронных СЖАТ на каждом из этих уровней.

# 2 СТРУКТУРНЫЕ МЕТОДЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ В МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ ЖЕЛЕЗНОДОРОЖНОЙ АВТОМАТИКИ

Структурные методы обеспечения безопасности являются наиболее распространенными и базируются на аппаратном и (или) программном резервировании элементов системы микропроцессорной централизации. Рассмотрим основные принципы обеспечения безопасности функционирования МИУС с использованием структурной избыточности.

## 2.1 Структуры безопасных МИУС

Структурные методы резервирования и контроля в МИУС, отвечающие требованиям безопасности, должны обеспечивать:

- независимость отказов в однотипных элементах функционально избыточных структур;
- защиту системы от сбоев и отказов, исключение возможности накопления отказов;
- контроль правильности функционирования программного обеспечения.

Всем этим требованиям удовлетворяют следующие основные типы безопасных структур МИУС, используемые при построении систем микропроцессорных централизаций.

### 2.1.1 Одноканальная система с одной программой и средствами внутреннего контроля и самотестирования

Одноканальная система с одной программой [2] может быть применена при организации достаточно полной проверки вычислительного канала с помощью самопроверяемых средств внутреннего контроля (ССВК) и наличии безопасных выходных схем (БВС) для включения управляемых объектов (УО) (рисунок 2.1).

Задачей самопроверяемой схемы внутреннего контроля является обнаружение неисправностей заданного класса в вычислительном канале и собственных неисправностей. При возникновении отказа ССВК формирует сигнал

*Y*, с помощью которого система может быть переведена в защитное состояние по входу ∅ (например, отключено питание), и (или) выходы ЭВМ *Z* отключаются от управляемых объектов УО с помощью БВС.

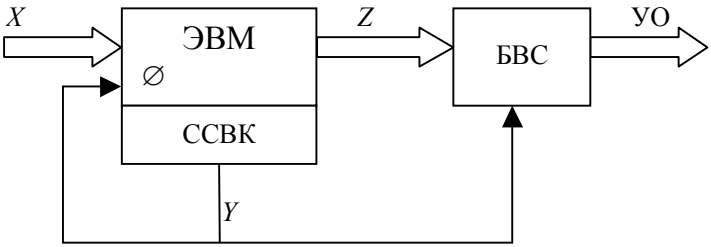



Рисунок 2.1 – Одноканальная система с одной программой

Безопасность данной структуры зависит от эффективности способа самопроверки. Обязательно должна присутствовать временная и информационная избыточность. Временная избыточность обеспечивается тестированием вычислительного канала. Тестовые программы должны выполняться перед каждым формированием управляющего воздействия и обеспечивать обнаружение всех одиночных отказов аппаратных средств.

Информационная избыточность реализуется кодированием информации, циркулирующей в системе и хранящейся в памяти, помехоустойчивыми кодами. Целесообразно применение самопроверяемого программного обеспечения.

Достоинством данной структуры является простота технической реализации. Недостатки: сложность определения критериев правильной работы системы для реализации ССВК; обнаружение отказов и сбоев только из заранее определенного множества дефектов; дополнительные затраты на разработку специального самопроверяемого программного обеспечения; снижение быстродействия из-за частого самотестирования; невысокая эксплуатационная готовность, т. к. любой отказ переводит систему в нерабочее защитное состояние.

В связи с этими недостатками данная структура получила ограниченное применение. Например, в системах микропроцессорной централизации она применяется только при условии использования в вычислительном канале безопасных ЭВМ.

### 2.1.2 Одноканальная система с диверситетными программами

Одноканальная система с диверситетными программами [2] использует две различные и независимые программы (П1 и П2) для реализации одних и тех же функций (рисунок 2.2). Результаты выполнения программ *Z1* и *Z2* сравниваются внешней безопасной схемой сравнения (БСС).

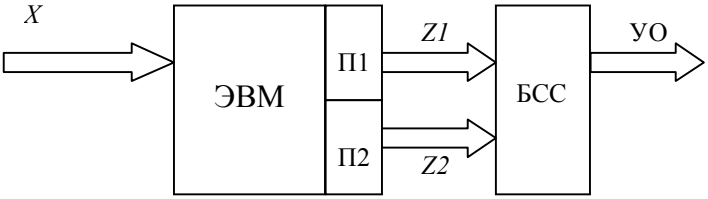



Рисунок 2.2 – Одноканальная система с диверситетными программами

Уровень безопасности зависит от степени различия (диверситета) двух программ и от интервала времени обращения к данным. Диверситет программ достигается использованием разных бригад программистов, алгоритмов, языков, подходов к программному обеспечению и методов описания спецификаций.

Наиболее известными формами программной диверситетной избыточности являются многоверсионное программирование, альтернативное программирование, избыточное кодирование информации, инверсионное повторение.

При многоверсионном программировании разными бригадами программистов разрабатываются две программы, которые реализуют одну и ту же задачу по разным алгоритмам. Система работает следующим образом: в фиксированные моменты времени считываются и запоминаются входные данные; выполняется обработка данных программой П1 и запоминание результатов; выполняется обработка данных программой П2 и сравнение результатов на совпадение; в случае совпадения происходит передача результирующих выходных воздействий на управляемые объекты, а в противном случае – блокировка системы.

При инверсионном повторении задача решается повторно, но по обратному алгоритму. Сначала реализуется программа П1, результаты выполнения которой служат входными данными для программы П2. Программа П2 решает обратную задачу (например, если П1 решает задачу $y=\sin x$, то П2 решает задачу $x=\arcsin y$). Результаты выполнения программы П2 должны

совпадать с входными данными программы П1, что и сравнивается внешней безопасной схемой сравнения.

Достоинствами данной структуры являются: простота технической реализации, обнаружение ошибок программного обеспечения. Недостатки: дополнительные затраты на разработку программного обеспечения; высокие требования к диверситету, т. к. отказы и сбои технических средств должны по разному влиять на результаты работы программ; снижение быстродействия из-за повторного выполнения программ; невысокая эксплуатационная готовность, т. к. любой отказ переводит систему в нерабочее защитное состояние.

Несмотря на эти недостатки, структуры с диверситетными программами нашли достаточно широкое применение как в одноканальных, так и двухканальных системах. Это обусловлено возможностью обнаружения ошибок в программном обеспечении, что обеспечить в недиверситетных структурах невозможно. Ограничивает применение таких структур высокая стоимость одной версии программного обеспечения, которая достигает 70 % общих затрат на разработку МИУС [3].

### 2.1.3 Дублированная система со слабыми связями

Дублированная система со слабыми связями [2] состоит из двух вычислительных каналов (рисунок 2.3), в которых процессоры и программы могут быть различными. Процессор ЭВМ1 реализует основные вычисления, а ЭВМ2 их проверяет. Для этого осуществляется обмен информацией по шине *W*. Синхронизация каналов не обязательна.

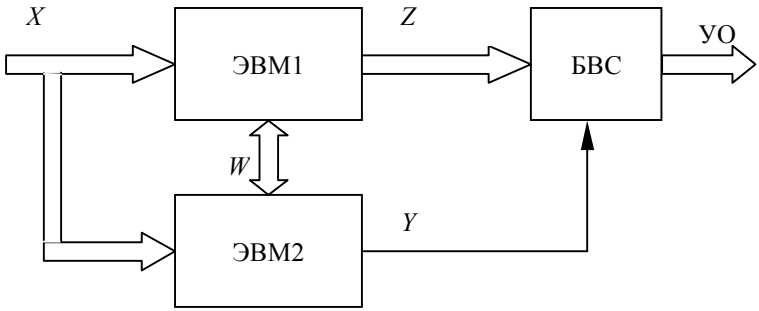



Рисунок 2.3 – Дублированная система со слабыми связями

Безопасность таких систем зависит от достоверности контроля функционирования вычислительных каналов. Контроль работы ЭВМ осуществляется либо за счет тестовых программ, либо за счет параллельных вычислений

одинаковыми или диверситетными программами и сравнения результатов. При обнаружении ошибки ЭВМ2 формирует сигнал *Y*, и выходы ЭВМ1 отключаются от УО с помощью БВС.

При организации параллельных вычислений по одинаковым алгоритмам возможно сравнение промежуточных результатов вычислений в контрольных точках. Сравнение результатов производится программно в ЭВМ2. Для обеспечения своевременного отключения отказавшего вычислительного канала необходимо, чтобы контролирующая ЭВМ выполняла вычисления не медленнее чем основная ЭВМ. Это достигается либо использованием процессора с более высоким быстродействием, либо использованием программ с более короткими алгоритмами вычислений. При этом необходимо обеспечить одновременное считывание исходных данных в оба вычислительных канала. Возможно использование идентичных вычислительных каналов и программного обеспечения.

Достоинства данной структуры: высокая гибкость; возможность использования диверситетных вычислительных каналов, что позволяет с высокой точностью обнаруживать отказы аппаратных средств и ошибки в программном обеспечении; отсутствие синхронизации каналов, что упрощает схемную реализацию и уменьшает вероятность возникновения одинаковых отказов и сбоев в обоих каналах.

Недостатки: высокие затраты на проектирование диверситетных вычислительных каналов и программного обеспечения; сложность выбора точек для контроля промежуточных вычислений, т. к. требуется обеспечить высокую достоверность контроля при минимальном количестве проверок; возможность накопления маскируемых отказов в обоих вычислительных каналах; необходимость организации контроля за правильностью программного сравнения результатов в ЭВМ2, т. к. отказы в ЭВМ2 не должны искажать результаты контроля; невысокая эксплуатационная готовность, т. к. любой отказ переводит систему в нерабочее защитное состояние.

На практике дублированные системы со слабыми связями применяются при построении различных систем железнодорожной автоматики, в том числе и систем централизации стрелок и сигналов. Например, система микропроцессорной централизации *Elektra*, разработанная фирмой *Alcatel* [4], состоит из двух идентичных ЭВМ, работающих с диверситетными программами.

### 2.1.4 Дублированная система с умеренными связями

Дублированная система с умеренными связями [2] включает в себя два одинаковых вычислительных канала с одинаковыми программами (рисунок

2.4). Это одна из наиболее распространенных на практике безопасных структур.

Работа обоих каналов синхронизирована. Синхронизация служит для одновременного считывания входных воздействий *X* и одновременной выдачи результатов *Z1* и *Z2*. Необходимость синхронизации обусловлена циклической непрерывной работой системы, когда даже при одинаковых технических средствах и одинаковых программах из-за разброса временных параметров со временем может произойти рассогласование работы каналов.

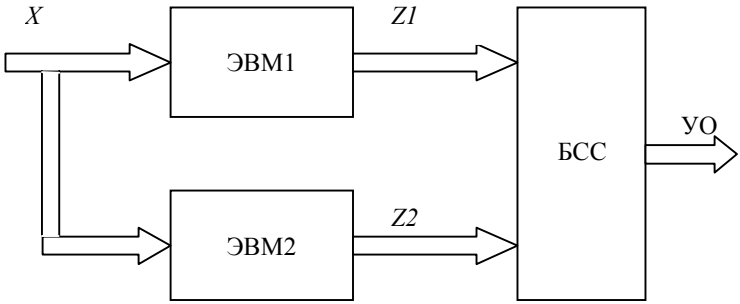



Рисунок 2.4 – Дублированная система с умеренными связями

Результаты обработки информации сравниваются на уровне выходов *Z1* и *Z2* с помощью БСС. При обнаружении рассогласования работы каналов БСС переводит свои выходы в защитное состояние и блокируется. Минимальная кратность необнаруживаемых отказов в системе равна двум – по одному отказу в каждом вычислительном канале, которые одинаковым образом искажают выходные сигналы *Z1* и *Z2*.

Одиночные отказы не опасны, если они искажают выходные сигналы и обнаруживаются БСС. В противном случае возможно накопление отказов. Для исключения накопления отказов вычислительные каналы снабжаются средствами самотестирования, т. е. реализуется временная избыточность. Тесты разрабатываются таким образом, чтобы любой отказ аппаратных средств на одной из тестовых последовательностей исказил значения выходных сигналов. Кратные независимые отказы должны обнаруживаться БСС.

Достоинства данной структуры: простота реализации; невысокая стоимость; высокая безопасность. Недостатки: возможность накопления маскируемых отказов в обоих вычислительных каналах; невозможность обнаружения ошибок в программном обеспечении, т. к. они одинаково проявляются в обоих каналах; невысокая эксплуатационная готовность, т. к. любой отказ переводит систему в нерабочее защитное состояние.

Структура получила широкое распространение ввиду невысокой стоимости и высокой безопасности. Применяется в основном в системах, не предъявляющих высокие требования к эксплуатационной готовности. Поэтому в системах микропроцессорных централизаций эта структура не нашла широкого применения и была вытеснена трехканальными структурами.

### 2.1.5 Дублированная система с сильными связями

Дублированная система с сильными связями [2] использует одинаковые программы в двух одинаковых вычислительных каналах, но в отличие от системы с умеренными связями контроль работы двух каналов осуществляется не только на уровне выходов, но и на уровне шин и памяти (рисунок 2.5).

Работа каналов синхронизирована. Синхронизация организуется либо по командам, либо по тактам. В наиболее сильном случае производится потактовая проверка совпадения сигналов *W1* и *W2* на внутренних контрольных точках (шинах) с помощью БСС1. При расхождении сигналов *W1* и *W2* БСС1 формирует сигнал ошибки *Y*. Сигнал *Y* воздействует на БСС2 и отключает УО, т. е. переводит оба канала в защитное состояние. Затем сигнал *Y*, поступая на вход ∅, блокирует работу двух ЭВМ.

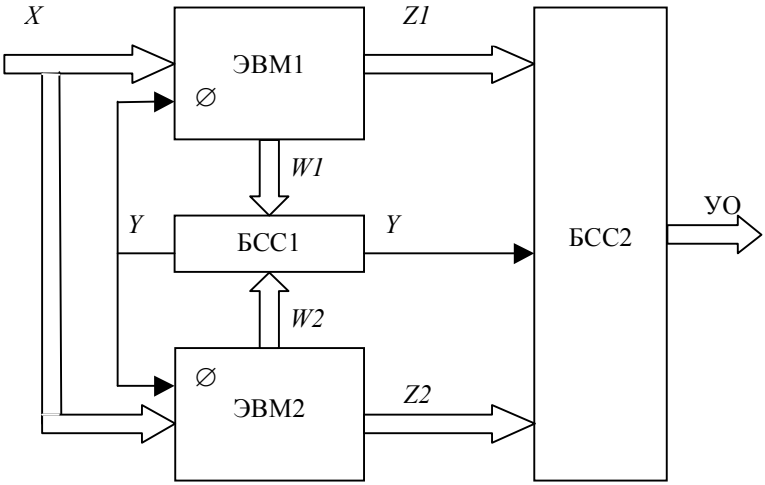



Рисунок 2.5 – Дублированная система с сильными связями

Структура обладает высоким уровнем безопасности, который зависит от вида и числа контролируемых разрядов (сигналы *W1* и *W2*). Одиночные отказы неопасны и должны обнаруживаться БСС1. Однако если множество входных воздействий *X* не обеспечивает необходимой глубины проверки каналов обработки информации, то возможно появление маскируемых отказов, т. е. отказов, которые не проявляются на данном промежутке времени в виде расхождения сигналов *W1* и *W2*. Накопление таких отказов может привести к опасному отказу системы. Это тем более актуально, что некоторые алгоритмы функционирования системы могут выполняться очень редко (раз в неделю или раз в месяц).

Достоинства данной структуры: невысокая стоимость; затраты на проектирование вычислительных каналов и программного обеспечения ниже, чем в других структурах; высокая глубина контроля отдельных функциональных узлов ЭВМ (процессора, памяти, портов ввода-вывода) при организации сравнения шин внутреннего интерфейса; высокая безопасность.

Недостатки: возможность накопления маскируемых отказов в редко используемых функциональных узлах вычислительных каналов; невозможность обнаружения ошибок в программном обеспечении, т. к. они одинаково проявляются в обоих каналах; необходимость обеспечения высокой надежности схемы синхронизации каналов; невысокая эксплуатационная готовность, т. к. любой отказ переводит систему в нерабочее защитное состояние.

Данная структура получила широкое распространение при проектировании систем железнодорожной автоматики, не предъявляющих высокие требования к эксплуатационной готовности. В системах микропроцессорной централизации для повышения эксплуатационной готовности применялось горячее резервирование еще одной двухканальной структурой.

Рассмотренные принципы обеспечения безопасности использовались в микропроцессорной централизации фирмы *Siemmens* первого поколения. Система строилась на базе защищенных от опасных отказов микропроцессорных блоков *SIMIS-C,* разработанных в 1985 году и имеющих двухканальную структуру с сильными связями. Для повышения эксплуатационной готовности блоки *SIMIS-C* дублировались. Один блок выполнял функции управления, второй в это время проводил самотестирование. При отказе основного блока управление передавалось резервному модулю. Однако в последнее время двухканальные системы начали вытесняться трехканальными мажорирующими структурами, имеющими лучшие показатели работы.

### 2.1.6 Дублированная система с сильными связями и внешним тестированием

Дублированная система с сильными связями и внешним тестированием [2] содержит в дополнение к структуре п. 2.1.5 генератор тестов (ГТ) и мультиплексор (МКС) и применяется, если множество входных воздействий *X* не обеспечивает необходимой глубины проверки каналов обработки информации (рисунок 2.6).

В этом случае в процессе рабочего функционирования периодически выделяются отрезки времени, в течение которых по сигналу *F*, поступающему на мультиплексор МКС, входные воздействия *X* отключаются от входов системы, и к последним подключаются выходы Т генератора тестов ГТ. Сигнал *F* также блокирует выходную схему сравнения БСС2, чтобы тестовые воздействия не прошли на объекты управления. Результаты тестирования обоих каналов сравниваются БСС1. При обнаружении ошибки система переводится в защитное состояние. После окончания тестирования сигнал *F* снимается, входные воздействия *X* подключаются к входам системы, и выходная схема сравнения включается в работу.

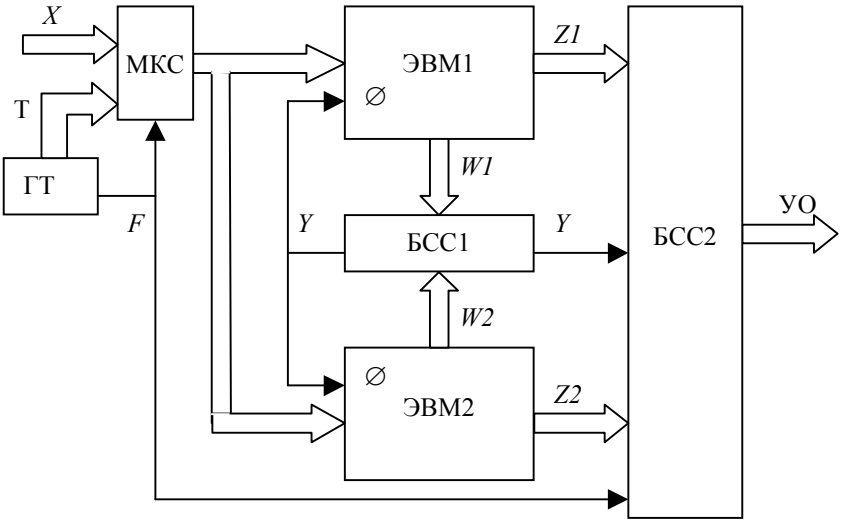



Рисунок 2.6 – Дублированная система с сильными связями и внешним тестированием

Данный принцип используется также тогда, когда система большую часть рабочего функционирования находится в ждущем режиме (при этом сигналы *X* длительное время не изменяются).

Безопасность такой структуры зависит от полноты тестов и времени, необходимого на тестирование, т. к. во время тестирования объект управления не контролируется. Поэтому применение дублированной системы с сильными связями и внешним тестированием оправдано в том случае, когда сумма времени тестирования и времени, необходимого на обработку входных воздействий, меньше времени реакции системы:

$$t_{\text{тест}} + t_{\text{обр}} \le t_{\text{р}} , \qquad (2.1)$$

где $t_{\text{тест}}$ – время, необходимое на тестирование;

$t_{\text{обр}}$ – время, необходимое на обработку входных воздействий;

$t_{\text{р}}$ – время реакции системы.

Достоинства данной структуры: высокая глубина контроля отдельных функциональных узлов ЭВМ (процессора, памяти, портов ввода-вывода); отсутствие маскируемых отказов; высокая безопасность. Недостатки: дополнительные затраты на разработку генератора тестов и тестовых воздействий; невозможность обнаружения ошибок в программном обеспечении; необходимость обеспечения высокой надежности схемы синхронизации каналов; невысокая эксплуатационная готовность, т. к. любой отказ переводит систему в нерабочее защитное состояние.

### 2.1.7 Самопроверяемая дублированная система

Самопроверяемая дублированная система [2] состоит из двух каналов (рисунок 2.7), построенных в виде самопроверяемых устройств. Каждый вычислительный канал снабжается самопроверяемой схемой внутреннего контроля ССВК, задачей которой является обнаружение неисправностей заданного класса в вычислительном канале и собственных неисправностей. Самопроверяемые схемы внутреннего контроля каждого канала при обнаружении ошибки вырабатывают сигнал *Y*, который отключает соответствующий вычислительный канал.

Выходные сигналы *Z1* и *Z2* сравниваются безопасной схемой сравнения БСС. При совпадении сигналов формируется управляющее воздействие на управляемый объект УО.

Сигналы контроля *W1* и *W2*, формируемые с помощью ССВК1 и ССВК2, сравниваются в ССВК3. При обнаружении ошибки ССВК3 вырабатывает сигнал *Y*, который переводит безопасную схему сравнения в защитное состояние.

Минимальная кратность необнаруживаемых отказов равна 4 – по два отказа в каждом канале, которые не обнаруживаются ССВК1 и ССВК2 и одинаковым образом искажают выходные сигналы *Z1* и *Z2*. Самоконтроль каналов может быть аппаратный и программный. Возможно использование независимых (диверситетных) программ в каждом процессоре.

Достоинства данной структуры: высокая глубина контроля вычислительных каналов; высокая безопасность; выделение отказавшего вычислительного канала. Недостатки: сложность определения критериев правильной работы системы для реализации ССВК; невозможность обнаружения ошибок в программном обеспечении при использовании одинаковых программ; сложность реализации системы, особенно при использовании диверситетных вычислительных каналов; невысокая эксплуатационная готовность, т. к. любой отказ переводит систему в нерабочее защитное состояние.

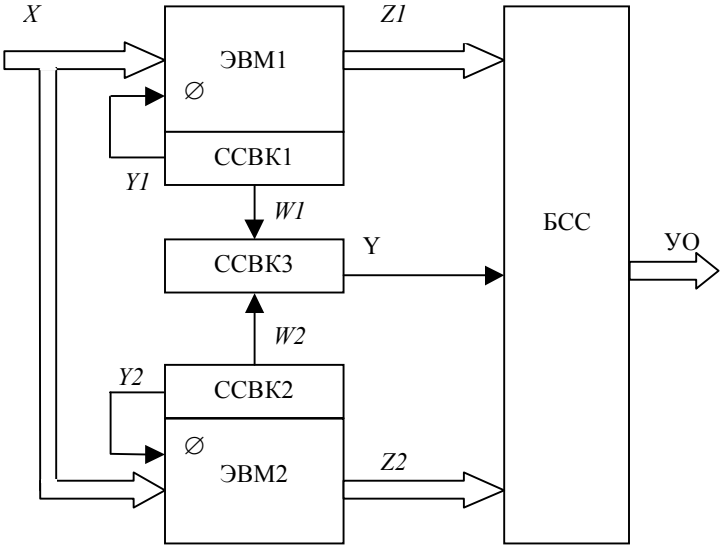



Рисунок 2.7 – Самопроверяемая дублированная система

Данная структура была использована в микропроцессорной централизации *E1A* фирмы *AEG* [5]. Все ЭВМ централизации построены на базе микропроцессорной системы с безопасными отказами *LOGISIRE C*. Эта система состоит из двух идентичных вычислительных каналов с одинаковым программным обеспечением. Каналы работают независимо друг от друга. Безопасность функционирования обеспечивается специальной операционной сис-

темой и защищенным от опасных отказов устройством контроля и отключения.

### 2.1.8 Мажоритарная система с умеренными связями

Мажоритарная система с умеренными связями [2] имеет три независимых канала обработки информации (рисунок 2.8). Каналы имеют идентичные аппаратные средства и программное обеспечение.

Работа каналов синхронизирована по входам и выходам, т. е. входная информация *X* должна считываться одновременно всеми каналами, а результаты обработки *Z1*, *Z2* и *Z3* одновременно должны поступать на входы безопасного мажоритарного элемента (БМЭ). БМЭ сравнивает сигналы *Z1*, *Z2* и *Z3* и формирует управляющее воздействие УО при совпадении хотя бы двух из трех входных сигналов. Кроме того, БМЭ определяет номер отказавшего канала и отключает его сигналами *Y1, Y2* или *Y3*. Безопасность такой структуры сравнима с безопасностью дублированной структуры с умеренными связями (см. рисунок 2.4), но отказоустойчивость ее выше.

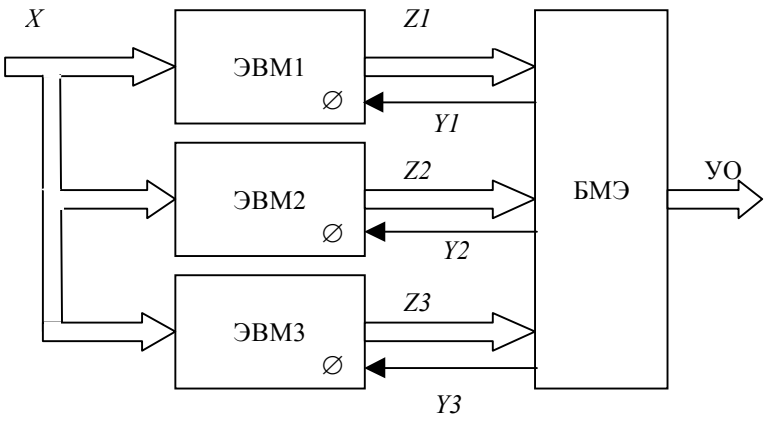



Рисунок 2.8 – Мажоритарная система с умеренными связями

Одиночные отказы неопасны, если они искажают выходные сигналы и маскируются БМЭ. В противном случае возможно накопление отказов. Для исключения накопления отказов вычислительные каналы снабжаются средствами самотестирования. Тесты разрабатываются таким образом, чтобы любой отказ аппаратных средств на одной из тестовых последовательностей

исказил значения выходных сигналов. Кратные независимые отказы должны обнаруживаться БМЭ.

Достоинства данной структуры: простота реализации; высокая безопасность; высокая эксплуатационная готовность, т. к. при отказе одного из каналов система продолжает выполнять свои функции. Недостатки: возможность накопления маскируемых отказов в вычислительных каналах; невозможность обнаружения ошибок в программном обеспечении, т. к. они одинаково проявляются во всех каналах.

Структура получила широкое распространение в виду относительно невысокой стоимости, высокой безопасности и эксплуатационной готовности.

### 2.1.9 Мажоритарная система с сильными связями

Мажоритарная система с сильными связями [2] использует одинаковые программы в трех одинаковых вычислительных каналах (рисунок 2.9), но в отличие от системы с умеренными связями, контроль работы трех каналов осуществляется не только на уровне выходов, но и на уровне шин и памяти.

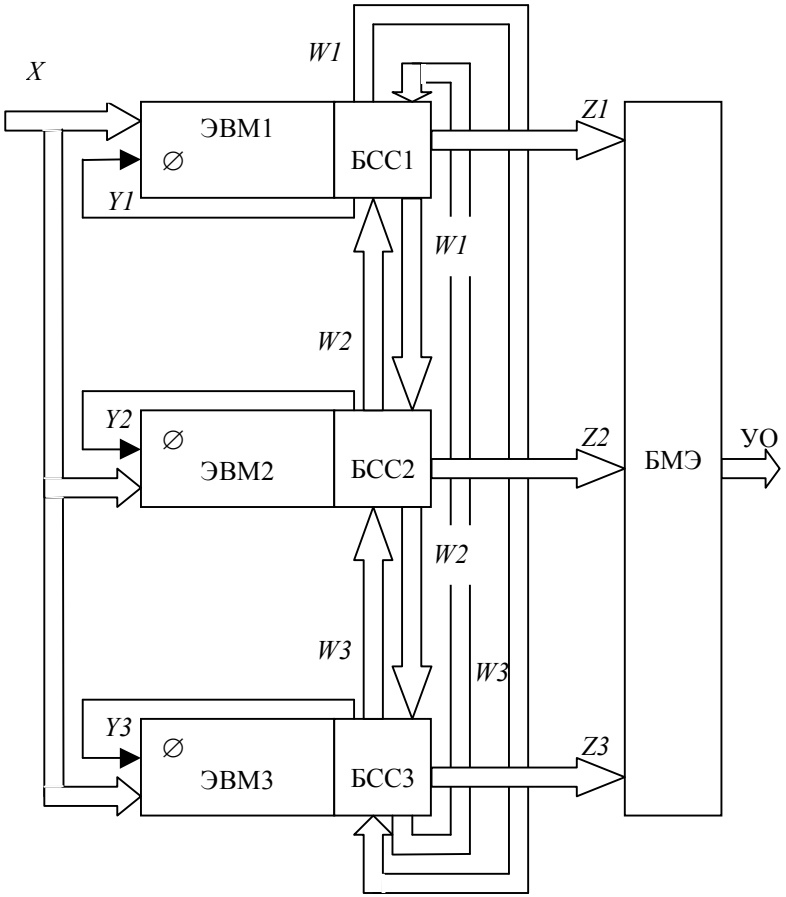



Рисунок 2.9 – Мажоритарная система с сильными связями

Безопасная схема сравнения (БСС) каждого канала производит потактовую проверку совпадения сигналов *W* двух других каналов с сигналом *W* своего канала на внутренних контрольных точках (шинах). При расхождении сигналов *W* БСС формирует сигнал ошибки *Y*. Сигнал *Y*, поступая на вход ∅, блокирует работу ЭВМ неисправного канала. Две другие ЭВМ в это время могут продолжать работу. Результаты обработки *Z1*, *Z2* и *Z3* сравниваются с помощью безопасного мажоритарного элемента (БМЭ).

Структура обладает высоким уровнем безопасности, который зависит от вида и числа контролируемых разрядов. Одиночные отказы не опасны и

должны обнаруживаться БСС. Не обнаруживаются ошибки в программном обеспечении, т. к. они одинаково проявляются во всех каналах. Безопасность такой структуры сравнима с безопасностью дублированной структуры с сильными связями (см. рисунок 2.5), но отказоустойчивость системы при этом выше.

Достоинства данной структуры: высокая глубина контроля отдельных функциональных узлов ЭВМ (процессора, памяти, портов ввода-вывода) при организации сравнения шин внутреннего интерфейса; высокая безопасность и эксплуатационная готовность. Недостатки: возможность накопления маскируемых отказов в редко используемых функциональных узлах вычислительных каналов; невозможность обнаружения ошибок в ПО; необходимость обеспечения высокой надежности схемы синхронизации каналов.

Данная структура получила широкое распространение в системах микропроцессорной централизации таких, как *SMILE* и *μSMILE* (Япония), микропроцессорной централизации фирмы *Siemmens* нового поколения.

### 2.1.10 Мажоритарная система с сильными связями и внешним тестированием

Мажоритарная система с сильными связями и внешним тестированием (рисунок 2.10) содержит в дополнение к структуре п. 2.1.9 генератор тестов (ГТ) и мультиплексор (МКС) и применяется, если множество входных воздействий $X$ не обеспечивает необходимой глубины проверки каналов обработки информации.

В этом случае в процессе рабочего функционирования периодически выделяются отрезки времени, в течение которых по сигналу $F$, поступающему на мультиплексор МКС, входные воздействия $X$ отключаются от входов системы, и к последним подключаются выходы Т генератора тестов ГТ. Сигнал $F$ также блокирует безопасный мажорирующий элемент БМЭ, чтобы тестовые воздействия не прошли на объекты управления. Результаты тестирования обоих каналов сравниваются БСС каждого канала. При обнаружении ошибки отказавший канал переводится в защитное состояние. После окончания тестирования сигнал $F$ снимается, входные воздействия $X$ подключаются к входам системы, и безопасный мажорирующий элемент включается в работу. Данный принцип используется также тогда, когда система большую часть рабочего функционирования находится в ждущем режиме (при этом сигналы $X$ длительное время не изменяются).

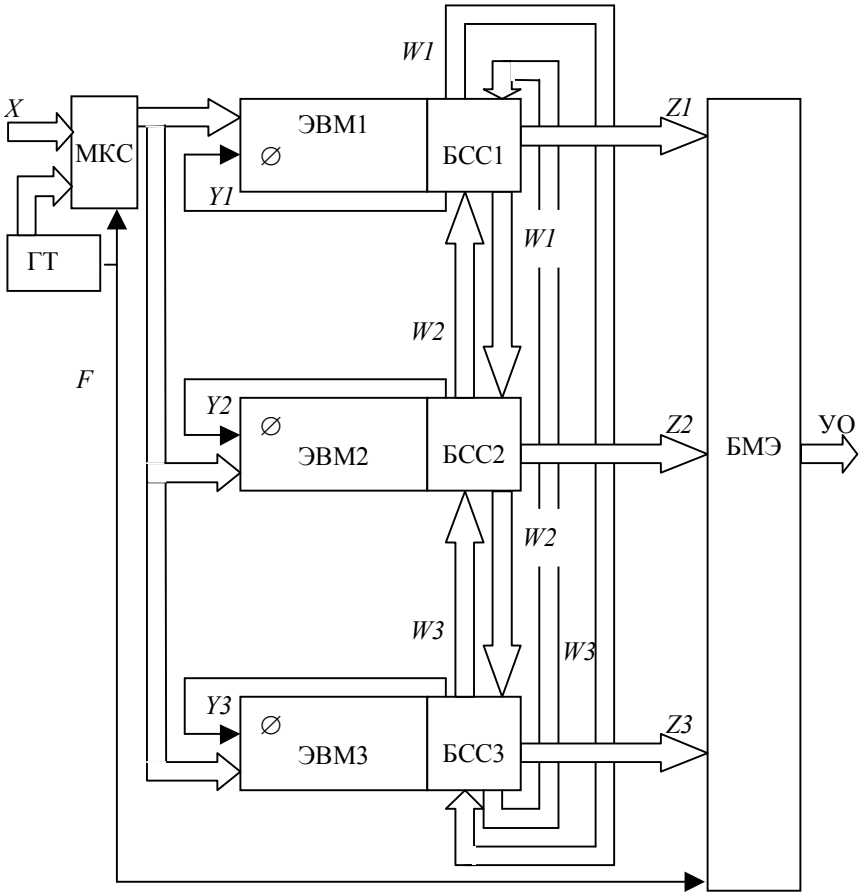



Рисунок 2.10 – Мажоритарная система с сильными связями и внешним тестированием

Безопасность такой структуры сравнима с безопасностью дублированной структуры с сильными связями и внешним тестированием (см. рисунок 2.6), но эксплуатационная готовность выше.

Безопасность такой структуры зависит от полноты тестов и времени, необходимого на тестирование, т. к. во время тестирования объект управления не контролируется.

Достоинства данной структуры: высокая глубина контроля отдельных функциональных узлов ЭВМ; отсутствие маскируемых отказов; высокая

безопасность и эксплуатационная готовность. Недостатки: дополнительные затраты на разработку генератора тестов и тестовых воздействий; невозможность обнаружения ошибок в программном обеспечении; необходимость обеспечения высокой надежности схемы синхронизации каналов.

Рассмотренные структуры могут использоваться в сочетании, дополняя друг друга. При этом базовыми обычно являются двухканальные, трехканальные или диверситетные структуры.

## 2.2 Реализация безопасных схем внутреннего контроля и сравнения

Во всех рассмотренных выше структурах безопасность обеспечивается специальными схемами контроля и сравнения. Данные схемы должны быть реализованы таким образом, чтобы исключить опасный отказ системы. Рассмотрим принципы построения ответственных схем контроля и сравнения в безопасных микропроцессорных структурах.

### 2.2.1 Принципы построения безопасных схем внутреннего контроля

Обмен информацией между отдельными узлами ЭВМ, входящих в состав МИУС, осуществляется через шины внутреннего интерфейса. Поэтому при контроле совпадения сигналов на этих шинах с заранее известными тестовыми значениями можно утверждать, что ЭВМ в процессе выполнения рабочих и тестовых алгоритмов функционирует без отказов. Таким образом можно контролировать исправность внутренних функциональных узлов ЭВМ. Устройство контроля и канал обработки информации выполняют в виде конструктивно законченного безопасного модуля.

Сравнение результатов обработки информации может производиться аппаратно или программно. При аппаратном сравнении информация, циркулирующая по внутренним шинам или хранящаяся в памяти ЭВМ, кодируется помехоустойчивыми кодами (равновесными, циклическими и др.). Безопасные схемы внутреннего контроля по тактам или в определенные моменты времени проверяют корректность кодированной информации.

В большинстве случаев устройство внутреннего контроля не определяет, какой узел отказал, а просто фиксирует отклонение в работе канала обработки информации. Первоначально, для того чтобы отличить сбой от отказа, устройство внутреннего контроля осуществляет перезапуск искаженного участка программы. При повторном обнаружении неравнозначности кодовых векторов на шине (в памяти) вычислительного канала осуществляется перевод ЭВМ в безопасное (выключенное) состояние. Причем отключение должно осуществляться необратимо даже в случае нового отказа в системе.

При выполнении рабочих алгоритмов МИУС некоторые элементы вычислительного канала могут использоваться с малой интенсивностью (например, некоторые области ОЗУ и ПЗУ). Поэтому для обеспечения большей глубины контроля и исключения возможности накопления отказов необходимо предусмотреть их циклическую тестовую проверку. Одним из видов такой проверки в паузах между эксплуатационными событиями является использование имитационных программ для тестового моделирования поездной обстановки на станции или перегоне. Таким образом, длительность периода контроля элементов вычислительного канала определяется рабочими и тестовыми алгоритмами системы.

Для контроля кодов с обнаружением ошибок используют самопроверяемые тестеры (СПТ). Они представляют собой кодовые детекторы, задача которых состоит в том, чтобы отличить кодовые комбинации, принадлежащие рассматриваемому коду, от остальных возможных комбинаций. СПТ реализуются в виде устройства с $n$ входами и двумя выходами $z_1$ и $z_2$.

Тестер обладает следующими свойствами:

• контролирует корректность входного вектора, т. е. выходы $z_1$ и $z_2$ принимают значения 1,0 или 0,1, если на входе тестера присутствует вектор рассматриваемого кода, и значения 0,0 или 1,1 в противном случае;

• выполняет самопроверку, т. е. для любой одиночной неисправности тестера существует входной вектор кода, на котором выходы $z_1$ и $z_2$ принимают значения 0,0 или 1,1.

На рисунке 2.11 показана схема тестера для четырехразрядного равновесного кода «2 из 4» (2/4-СПТ). На входы $x_1$, $x_2$, $x_3$ и $x_4$ подаются четырехразрядные двоичные коды. В таблице 2.1 представлено преобразование кода самопроверяемым тестером.

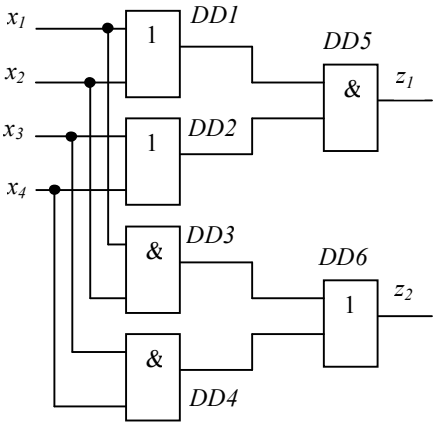



Рисунок 2.11 – Схема тестера для четырехразрядного равновесного кода 2/4-СПТ

Правильным считается код, вес которого равен двум (две единицы в коде). Свойство самопроверки заключается в том, что для любой одиночной неисправности элементов тестера можно найти такую кодовую комбинацию, которая установит выходы тестера в состояние 0,0 или 1,1. На-

пример, отказ «1→0» на выходе элемента *DD1* обнаруживается кодом «0101», т. к. на выходе установится значение 0,0.

Тестеры характеризуются двумя оценками: сложностью, которая равна суммарному числу входов логических элементов, принадлежащих структуре тестера, и длиной проверяющего теста, которая равна числу слов кода, подача которых на вход тестера обеспечивает обнаружение всех одиночных неисправностей. Для тестера 2/4-СПТ проверяющий тест равен T={0101, 0011, 1100}.

*Таблица 2.1*

| Четырехразрядный равновесный код | | | | Выходы | | Результат контроля |
|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $z_1$ | $z_2$ | |
| 0 | 0 | 0 | 0 | 0 | 0 | Ошибка, код с меньшим весом |
| 0 | 0 | 0 | 1 | 0 | 0 | Ошибка, код с меньшим весом |
| 0 | 0 | 1 | 0 | 0 | 0 | Ошибка, код с меньшим весом |
| 0 | 0 | 1 | 1 | 0 | 1 | Верный код |
| 0 | 1 | 0 | 0 | 0 | 0 | Ошибка, код с меньшим весом |
| 0 | 1 | 0 | 1 | 1 | 0 | Верный код |
| 0 | 1 | 1 | 0 | 1 | 0 | Верный код |
| 0 | 1 | 1 | 1 | 1 | 1 | Ошибка, код с большим весом |
| 1 | 0 | 0 | 0 | 0 | 0 | Ошибка, код с меньшим весом |
| 1 | 0 | 0 | 1 | 1 | 0 | Верный код |
| 1 | 0 | 1 | 0 | 1 | 0 | Верный код |
| 1 | 0 | 1 | 1 | 1 | 1 | Ошибка, код с большим весом |
| 1 | 1 | 0 | 0 | 0 | 1 | Верный код |
| 1 | 1 | 0 | 1 | 1 | 1 | Ошибка, код с большим весом |
| 1 | 1 | 1 | 0 | 1 | 1 | Ошибка, код с большим весом |
| 1 | 1 | 1 | 1 | 1 | 1 | Ошибка, код с большим весом |

Виды отказов, обнаруживаемых проверяющим тестом, приведены в таблице 2.2.

*Таблица 2.2*

| Неисправности | | Проверяющий код | Значение при исправной работе | Значение при наличии неисправности |
|---|---|---|---|---|
| вид | элемент | | | |
| «1→0» | *DD1* | 0101 | 10 | 00 |
| «0→1» | *DD1* | 0011 | 01 | 11 |
| «1→0» | *DD2* | 0101 | 10 | 00 |
| «0→1» | *DD2* | 1100 | 01 | 11 |
| «1→0» | *DD3* | 1100 | 01 | 00 |
| «0→1» | *DD3* | 0101 | 10 | 11 |
| «1→0» | *DD4* | 0011 | 01 | 00 |
| «0→1» | *DD4* | 0101 | 10 | 11 |

| | | | | |
|---|---|---|---|---|
| «1→0» | *DD5* | 0101 | 10 | 00 |
| «0→1» | *DD5* | 1100 | 01 | 11 |
| «1→0» | *DD6* | 1100 | 01 | 00 |
| «0→1» | *DD6* | 0101 | 10 | 11 |

Самопроверяемые тестеры можно использовать для контроля правильности работы как одноканальных структур, так и многоканальных.

### 2.2.2 Реализация безопасных схем внутреннего контроля в одноканальной структуре

В одноканальной структуре невозможно обеспечить передачу по шинам только информации, кодированной равновесным кодом. Поэтому контролирующее устройство должно подключаться не непосредственно к шинам микропроцессора, а к буферному элементу (регистру или порту вывода). Микропроцессор в определенные такты времени (контрольные точки) записывает кодированную информацию в буфер, а устройство контроля проверяет корректность кода.

Для исключения накопления ошибок за время диагностирования $t_д$ должно быть обеспечено поступление контрольной информации (проверяющего теста), достаточной для проверки исправности как вычислительного канала, так и устройства контроля. Т. е. коды должны быть различны и проверять все возможные одиночные отказы в устройстве контроля. Время $t_д$ и количество контролируемых разрядов выбираются исходя из требуемых показателей безопасности системы.

На рисунке 2.12 приведена функциональная схема восьмиразрядного устройства контроля вычислительного канала.

При правильной работе вычислительного канала на выходах самопроверяемых тестеров 2/4-СПТ$_1$, 2/4-СПТ$_2$ и 2/4-СПТ$_3$ присутствует статический парафазный сигнал (01 или 10). Этот сигнал поступает на входы 2/4-СПТ$_4$. На другие входы 2/4-СПТ$_4$ поступают парафазные импульсы от парафазного тактового генератора (ПТГ). При этом на выходе 2/4-СПТ$_4$ формируется импульсный парафазный сигнал. Импульсный парафазный сигнал через парафазные триггеры ПТ$_1$ и ПТ$_2$ поступает на самопроверяемую схему включения реле ССВР, которая поддерживает реле Р во включенном состоянии. Своими контактами реле Р коммутирует цепи питания микропроцессорной системы, поддерживая ее в рабочем состоянии.

При появлении запрещенной кодовой комбинации в контрольном регистре вычислительного канала на выходах контрольных схем 2/4-СПТ появляется непарафазный сигнал, который регистрируется фиксирующим элементом (ФЭ). Для того, чтобы отличать сбои и отказы аппаратуры, ФЭ содержит два последовательно соединенных парафазных триггера ПТ$_1$ и ПТ$_2$.

При первоначальном нарушении парафазности на выходах контрольных схем оба триггера блокируются, и в микропроцессор поступает запрос прерывания. По этому сигналу в микропроцессоре осуществляется возврат в программе на несколько шагов назад (рестарт), формируется сигнал восстановления $ПТ_1$, и искаженный участок программы повторяется вновь. Время формирования сигнала восстановления должно быть меньше времени отпускания реле Р, чтобы реле не успело отключить питание системы.

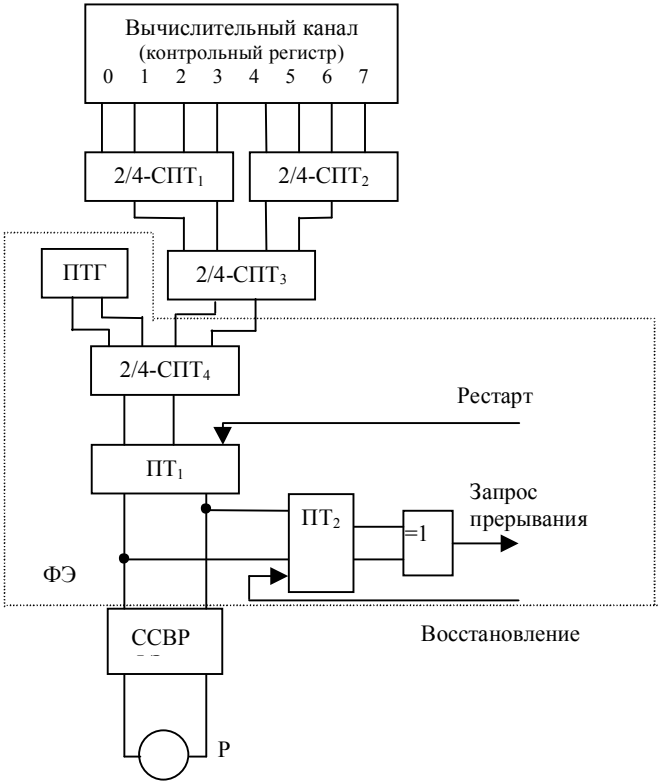



Рисунок 2.12 – Функциональная схема восьмиразрядного устройства контроля вычислительного канала

Если снова фиксируется нарушение выполнения программы, то $ПТ_1$ окончательно блокируется, на вход ССВР поступает не парафазный сигнал, и реле Р своими контактами выключает питание микропроцессора модуля, т. е. обеспечивается защитное состояние МИУС. При отсутствии повторного

сбоя, т. е. при полном прохождении первоначально искаженного программного блока, микропроцессор формирует сигнал восстановления $ПТ_2$.

Парафазный триггер (ПТ) представляет собой устройство (рисунок 2.13), имеющее два парафазных информационных входа $T^0$ и $T^1$ и два выхода $Q^0$ и $Q^1$.

Парафазный триггер обладает следующими свойствами:

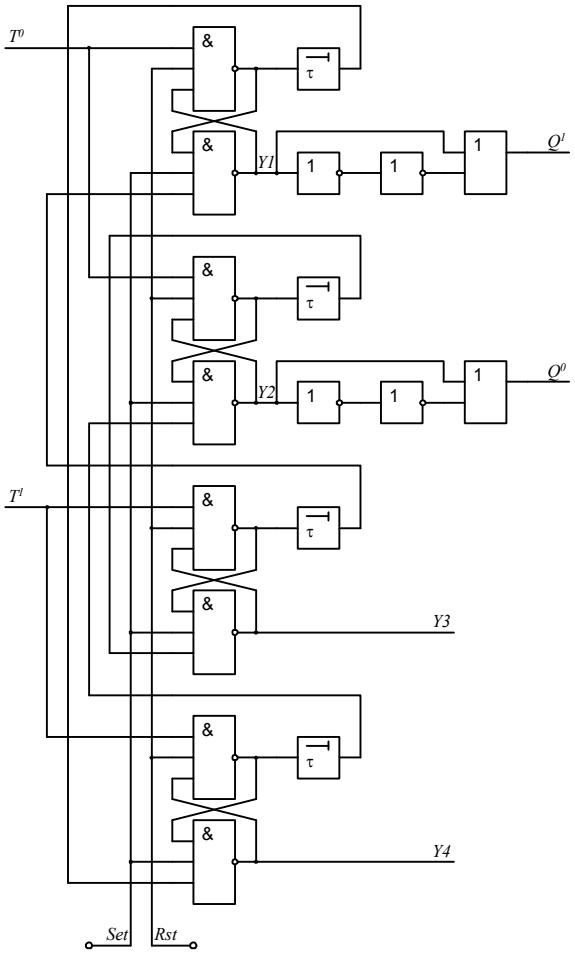
- если на вход парафазного триггера поступает парафазный сигнал и сама схема ПТ исправна, то на его выходе также присутствует парафазный сигнал;
- если на вход ПТ в любом такте его работы поступают одинаковые сигналы, то схема ПТ блокируется в защитном состоянии, и в том же такте на выходе устанавливаются одинаковые сигналы (0,0 или 1,1), которые сохраняются во всех последующих тактах работы независимо от состояния входов;

• при возникновении в схеме ПТ одиночных неисправностей схема также блокируется в защитном состоянии;

Рисунок 2.13 – Парафазный триггер

• вывод схемы из защитного состояния возможен только по цепям установки (*SET* или *RST*).

Парафазный триггер состоит из четырех бистабильных ячеек памяти *Y1, Y2, Y3* и *Y4*. При кратковременной подаче логического нуля на вход *RST* и наличии сигналов $T^1T^0$=01 схема переходит в устойчивое состояние 0110 (со-




стояние «0» триггера). При подаче логического нуля на вход *SET* и наличии

сигналов $T^1T^0$=01 схема переходит в устойчивое состояние 1001 (состояние «1» триггера). Полный цикл работы триггера происходит при поступлении входной последовательности сигналов $T^1T^0$ вида 01,10,01,10,01. При этом схема последовательно проходит все свои состояния 0110 → 1010 → 1001 → 0101 → 0110.

Примеры реализации ССВР будут рассмотрены в п. 4.2 «Устройства включения исполнительных реле».

### 2.2.3 Реализация безопасных схем сравнения в многоканальных структурах

Безопасные схемы сравнения в многоканальных структурах МИУС выполняют сравнение соответствующих сигналов на шинах (в памяти) каналов между собой. На рисунке 2.14 приведена обобщенная структура дублированного микропроцессорного модуля. Шины внутреннего интерфейса контролируются безопасной схемой сравнения с несимметричной характеристикой отказов (БСС1).

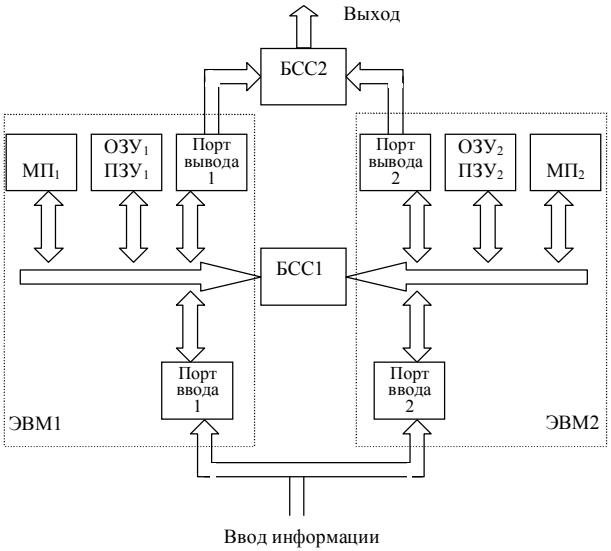



Рисунок 2.14 – Обобщенная структура дублированного микропроцессорного модуля

Выходная информация на внешнем интерфейсе формируется схемой сравнения БСС2, контролирующей идентичность состояния выходов обеих ЭВМ.

Для реализации безопасных схем сравнения можно воспользоваться свойствами самопроверяемых тестеров для равновесных кодов. Пример

функциональной схемы устройства сравнения шин дублированной системы приведен на рисунке 2.15.

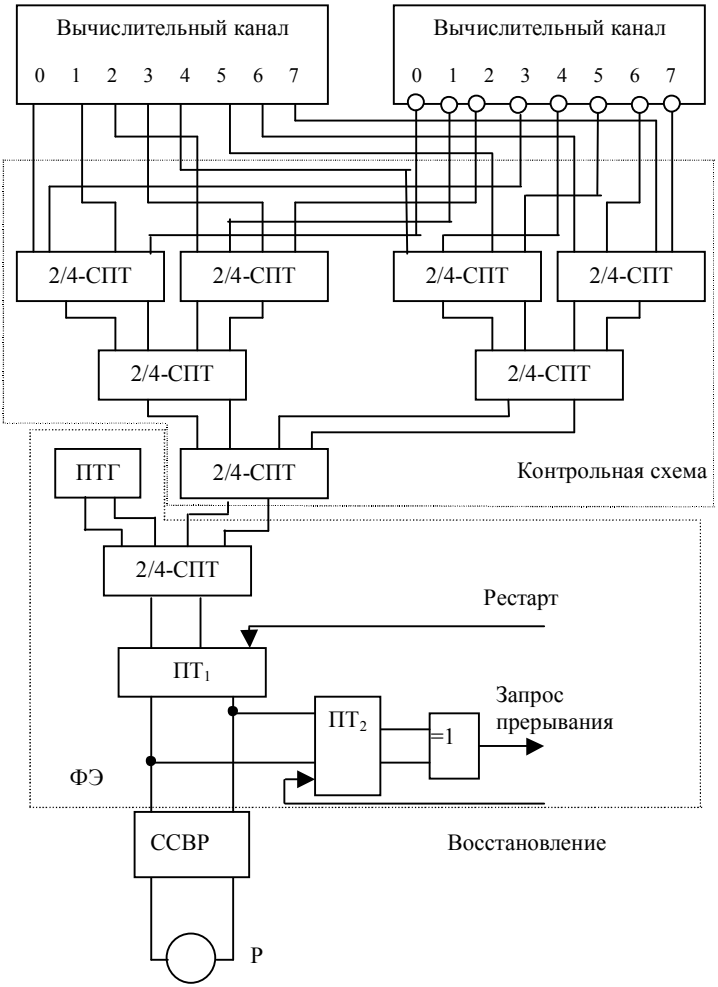



Рисунок 2.15 – Функциональная схема устройства сравнения шин дублированной системы

Сигналы от второго микропроцессора поступают на устройство сравнения в инверсном виде. На каждый 2/4-СПТ поступают по два соответствующих сигнала с каждого канала. Таким образом формируется равновесный код.

При нарушении согласованной работы микропроцессоров обоих каналов на выходах контрольной схемы появляется непарафазный сигнал, который

регистрируется фиксирующим элементом (ФЭ). Структура ФЭ аналогична представленной на рисунке 2.12.

Устройство контроля шин трехканальной структуры может быть также выполнено на основе самопроверяемых тестеров (рисунок 2.16). Сигналы на шинах микропроцессоров попарно сравниваются, так же как и в дублированной структуре, с помощью тестеров 2/4-СПТ.

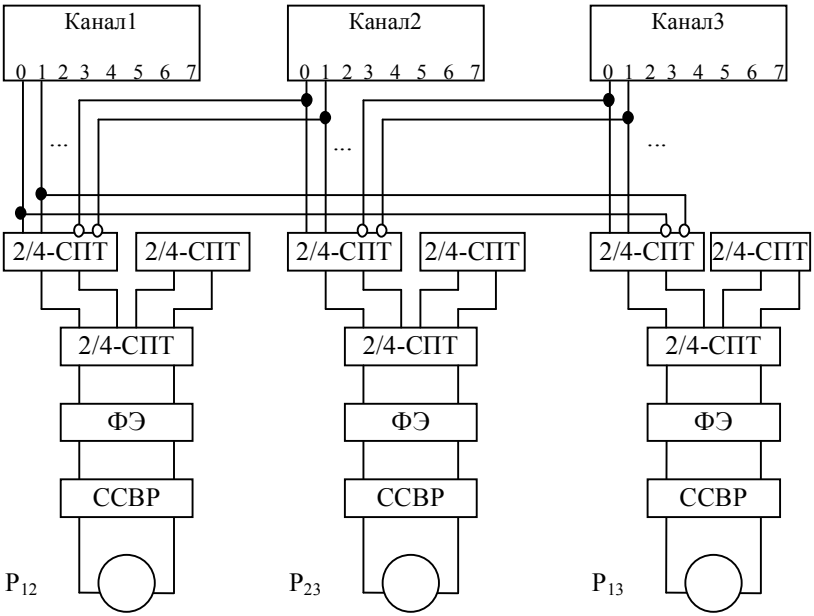



Рисунок 2.16 – Устройство контроля шин трехканальной структуры

При отказе одного из микропроцессоров выключаются два из трех контрольных реле. Например, при отказе первого канала отключатся реле $P_{12}$ и $P_{13}$, т. к. реле $P_{12}$ контролирует синхронную работу первого и второго каналов, а реле $P_{13}$ – первого и третьего каналов. С помощью контактов контрольных реле осуществляется дешифрация номера неисправного канала и его отключение (рисунок 2.17).

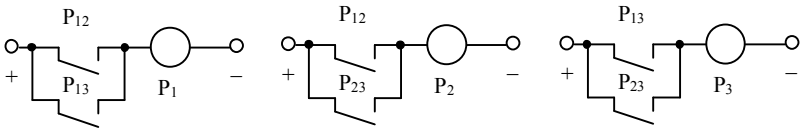


Рисунок 2.17 – Схема дешифрации номера неисправного канала

Выключение двух контрольных реле влечет за собой выключение одного из питающих реле, которое своими контактами отключает питание неисправного канала. Система при этом деградирует до двухканальной системы, сохраняя свою работоспособность. При восстановлении неисправного канала, он может быть включен в работу только при участии человека. Момент включения восстановленного канала должен быть синхронизирован с работой остальных каналов.

Если до восстановления неисправного канала произойдет еще один отказ, то произойдет рассогласование работы двух оставшихся каналов. Это приведет к выключению последнего контрольного реле, которое выключит оба оставшихся питающих реле. Все три канала будут выключены и система перейдет в защитное состояние.

Сократить число элементов и значительно повысить надежность устройства контроля микропроцессорных модулей можно за счет сравнения кодовых последовательностей на шинах не в параллельном виде, а в последовательном. С этой целью для мультиплексирования сигналов на шинах микропроцессора можно использовать универсальные сдвиговые регистры (рисунок 2.18).

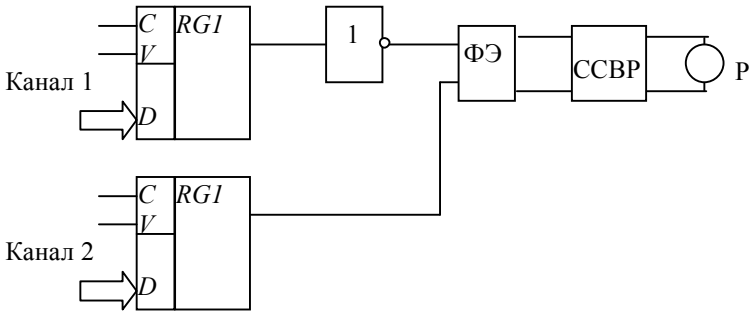



Рисунок 2.18 – Последовательное устройство контроля дублированной структуры

В один и тот же момент времени производится запись информации с шин вычислительных каналов в соответствующие сдвиговые регистры. Затем по импульсам с тактового генератора производится поразрядное сравнение данных в обоих регистрах. При обнаружении рассогласования данных блокируется ФЭ и выключается питающее реле.

Структурная схема устройства контроля шин мажоритарно-резервированных микропроцессорных моделей приведена на рисунке 2.19. В данном случае сигналы каждого канала сравниваются с выходом мажоритарного элемента «2 из 3». Сигнал на выходе мажоритарного элемента определяется большинством входных сигналов. В устройстве контроля не требуется дешифратор неисправного канала, т. к. контрольное реле подключено к соответствующему каналу обработки информации. Поэтому, при отказе вычислительного канала, выключается соответствующее реле.

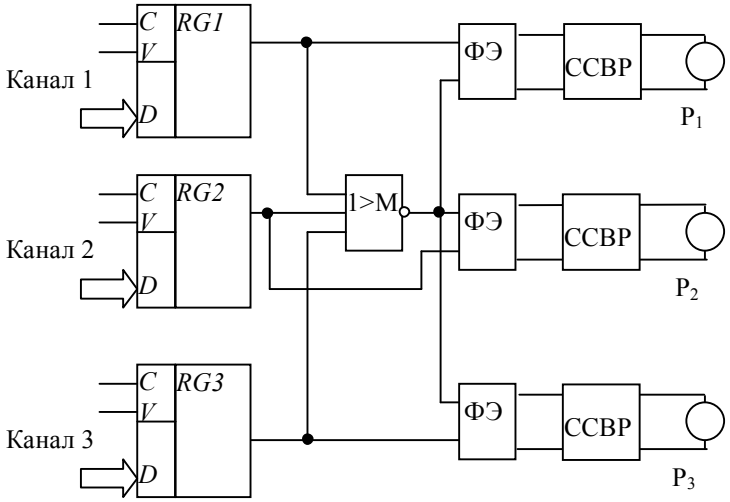



Рисунок 2.19 – Последовательное устройство контроля мажоритарной структуры

Для повышения отказоустойчивости устройства контроля мажоритарный элемент может быть выполнен резервированным, как это показано на рисунке 2.20.

Последовательные устройства контроля для обеспечения высокой достоверности должны выполнять операции сравнения сигналов в течение одного такта (одной команды) работы микропроцессора в зависимости от степени синхронизации каналов.

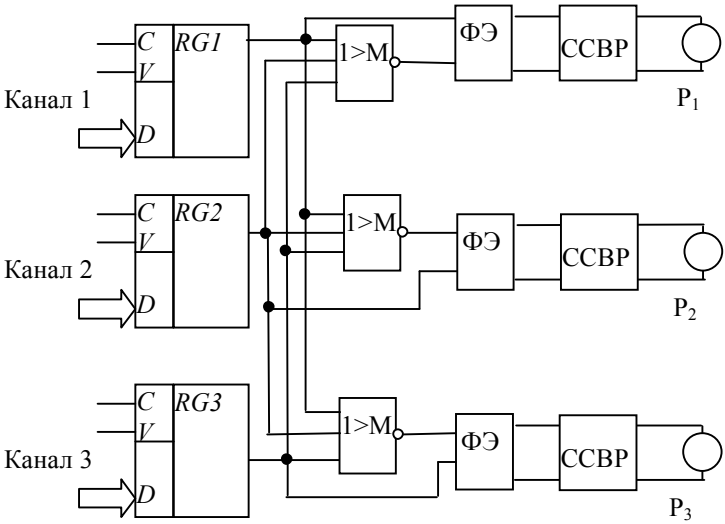



Рисунок 2.20 – Последовательное устройство контроля мажоритарной структуры с резервированным мажоритарным элементом

Таким образом, во второй главе рассмотрены основные структуры безопасных микропроцессорных систем, их достоинства и недостатки. Приведены типовые решения по организации аппаратного контроля и способам локализации отказов в этих структурах. В следующей главе рассмотрим принципы построения безопасных логических элементов, использующихся на аппаратном уровне защиты для создания контрольных элементов МИУС.

# 3 БЕЗОПАСНЫЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

Логические элементы играют важную роль при синтезе безопасных дискретных систем. От их свойств и особенностей зависит концепция обеспечения безопасности. Рассмотрим основные принципы построения безопасных логических элементов.

## 3.1 Принципы построения безопасных логических элементов

Логические элементы (ЛЭ) реализуют одну из элементарных функций алгебры логики (ФАЛ): «НЕ», «И», «ИЛИ» и некоторые другие. ЛЭ имеют один выход с двумя возможными состояниями: «0» и «1». В зависимости от входных данных и алгоритма функционирования исправный элемент осуществляет функциональные переходы $0 \rightarrow 1$ и $1 \rightarrow 0$. Если же переход из одного состояния в другое происходит в результате отказа, то такой переход называют ложным или нефункциональным.

Если интенсивность отказов ЛЭ, вызывающих один из двух ложных (нефункциональных) переходов выходного сигнала ($0 \rightarrow 1$ или $1 \rightarrow 0$), выше интенсивности отказов, вызывающих другой переход, на порядок и более, то такие ЛЭ называют элементами с несимметричными отказами.

Элемент с несимметричными отказами, у которого интенсивность возникновения менее вероятного вида отказов не более предельного значения при заданном уровне безопасности, называют безопасным элементом.

Элементы с несимметричным отказом разрабатываются специально для построения безопасных систем. Несимметричность отказов ЛЭ достигается следующими методами:

- специальным физическим представлением логических сигналов;
- резервированием деталей и узлов;
- импульсным кодированием сигналов;
- использованием генераторных и резонансных режимов работы;
- гальванической развязкой входных и выходных цепей;
- избыточным кодированием внутренней и внешней информации;
- специальными конструктивными мерами.

Неисправности, возникновение которых исключено природными законами, свойствами использованных материалов, конструкцией и технологией изготовления компонентов и изделий называются недопустимыми. Например, реле I класса надежности не может не отпустить якорь под действием силы тяжести, серебряно-угольные контакты не свариваются благодаря свойству материалов, короткое замыкание первичной и вторичной обмоток трансформатора исключается при двухкамерном каркасе катушки, изменение частоты резонансного контура, при обрыве конденсатора, исключается применением конденсатора специальной конструкции с четырьмя выводами.

В нормативных документах (EN, ОСТ РФ, РД РБ БЧ) установлено предельное значение интенсивности опасных отказов ЛЭ: $\lambda_{доп}=1\cdot10^{-11}$ 1/ч.

Интенсивность недопустимых неисправностей принята $\lambda_{недоп}=1\cdot10^{-13}$ 1/ч.

Неисправности, интенсивность возникновения которых ниже этого предела, при разработке ЛЭ и анализе безопасности можно не учитывать.

У микроэлектронной элементной базы (диодов, транзисторов, БИС и т. д.), как правило, нет недопустимых неисправностей. Они являются элементами с симметричными отказами. Интенсивность отказов типа «короткое замыкание» и «обрыв» имеет одинаковый порядок и на 4-5 порядков выше нормы $\lambda_{доп}$. Поэтому в безопасных ЛЭ реализуется принцип безопасного поведения при отказе. У таких элементов допустимый ложный переход ($1\rightarrow0$) назван защитным отказом, а недопустимый ($0\rightarrow1$), названный опасным отказом, отсутствует.

Защитные отказы таких элементов могут происходить с большой интенсивностью ($\lambda_з=1\cdot10^{-3}$ – $1\cdot10^{-6}$ 1/ч). Однако они отвечают требованиям безопасности, т. к. их опасные отказы возникают с интенсивностью ниже нормы. Реализация безопасного последействия отказов основана на самоконтроле и контроле ЛЭ.

Если ЛЭ построен так, что защитный переход на его выходе происходит немедленно после активизации неисправности без использования контрольных средств, то ЛЭ называется логическим элементом с безопасным поведением при отказе (*fail safe* ЛЭ).

Логический элемент построен на принципе квазибезопасности (*quasi fail safe* ЛЭ), если отказы в его внутренней структуре обнаруживаются по значению выходного сигнала элемента с использованием избыточных контрольных средств, реализующих принятый критерий исправности и переключающих ЛЭ в защитное состояние немедленно после обнаружения отказа. В квазибезопасных ЛЭ ответственность за безопасность «переносится» с самого ЛЭ на контрольный элемент [6].

Рассмотрим примеры реализаций безопасных логических элементов.

### 3.2 Декодеры сигналов логических переменных

В традиционных дискретных устройствах положительной логики логический «0» представляется низким (нулевым), а логическая «1» – высоким устойчивым потенциалом. Преимущества такого представления логических сигналов: четкость физических процессов, естественность работы схемы, простота алгоритма обработки информации и экономичность технических решений.

Чтобы отличить нормальное функционирование устройства от его работы в момент отказа и зафиксировать отказ, необходимо ввести признак отказа, который соответствует не традиционно принятым низкому и высокому потенциалам, а другим видам сигнала. Для этих целей вводят кодирование сигналов логических переменных и кодированную обработку информации.

Рассмотрим кодирование входных сигналов на примере ЛЭ, который включает лампу зеленого света (рисунок 3.1, *а*).

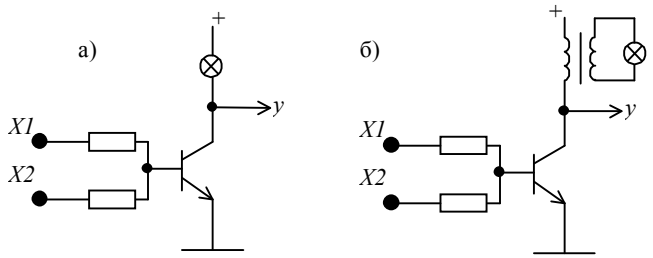



Рисунок 3.1 – Примеры реализации резисторно-транзисторных элементов

На входы ЛЭ поступают некодированные сигналы *X1* и *X2*. Лампа загорается, если *X1*=1 и *X2*=1. Но лампа горит и при неисправности «пробой эмиттер-коллектор» транзистора, что соответствует опасному отказу.

Использование простейшего однофазного кодирования, при котором логическая «1» представляется импульсами постоянного тока, а логический «0» – устойчивым сигналом, позволяет преобразовать этот отказ из опасного в защитный (рисунок 3.1, *б*). Лампа горит при наличии импульсных сигналов 1 на входах *X1* и *X2*. Устойчивый сигнал на входе (0 или 1) приводит к тому, что лампа гаснет. Лампа гаснет также при обрыве и коротком замыкании цепи «эмиттер-коллектор» транзистора.

При необходимости можно перейти к парафазному кодированию, при котором для представления значения логической переменной выделяются две фазы (пространственная и временна́я). Сигнал $X$ передается с помощью единичной $x$ и нулевой $\overline{x}$ фаз, т. е. $X = x\overline{x}$. Сигнал логического «0» кодиру-

ется в этом случае как «01», а сигнал логической «1» – «10». Остальные сигналы – «00» или «11» – соответствуют защитному состоянию.

В схемах ЛЭ с кодированными логическими переменными применяют входные преобразователи (декодеры). Однофазный декодер преобразует импульсный такт или гармонический сигнал в постоянное напряжение $U$, а устойчивый сигнал – в нулевой потенциал, т. е.

$$y = \begin{cases} y = 1, & \text{если } x = T \text{ (импульсный такт)}; \\ y = 0, & \text{если } x = C \text{ (постоянная)}. \end{cases}$$

Парафазный декодер преобразует прямую фазу в постоянное напряжение $U$, а обратную – в нулевой потенциал, т. е.

$$y = \begin{cases} y = 1, & \text{если } x = T \text{ (такт)}; \\ y = 0, & \text{если } x = \overline{T} \text{ (противотакт)}. \end{cases}$$

Декодеры являются элементами без опасных отказов. Критерий безопасности декодера – отсутствие набора допустимых неисправностей в структуре его схемы, в результате которого может произойти ложный переход $0 \rightarrow 1$ при наличии логического «0» на его входе.

Для выпрямления однофазного сигнала часто применяют выпрямительные схемы с трансформаторами (рисунок 3.2 *а, б*). Трансформатор обладает исключительным для безопасности свойством – не переносить сигнал в следующий каскад схемы, если сигнал в первичной обмотке становится постоянным.

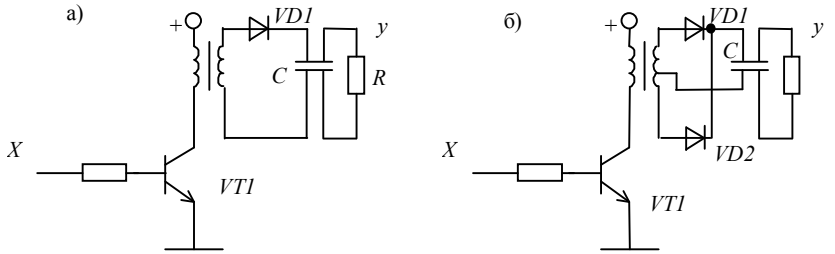



Рисунок 3.2 – Схемы декодирования однофазных сигналов

Пульсации выходного сигнала должны быть минимальны, в противном случае они могут быть восприняты как импульсный такт. Поэтому на выходе для сглаживания выпрямленного напряжения включают четырехвыводной конденсатор. Его обрывы исключают формирование выходного сигнала.

Наличие трансформатора в рассмотренных схемах имеет свои недостатки. Это связано с тем, что технологически трудно и экономически невыгодно

применять трансформатор в микросхемах с высокой степенью интеграции. Поэтому более широкое применение нашли конденсаторные декодеры.

Типовые схемы конденсаторных декодеров, отличающиеся знаком напряжения питания и выходного сигнала, представлены на рисунке 3.3 (*а, б*).

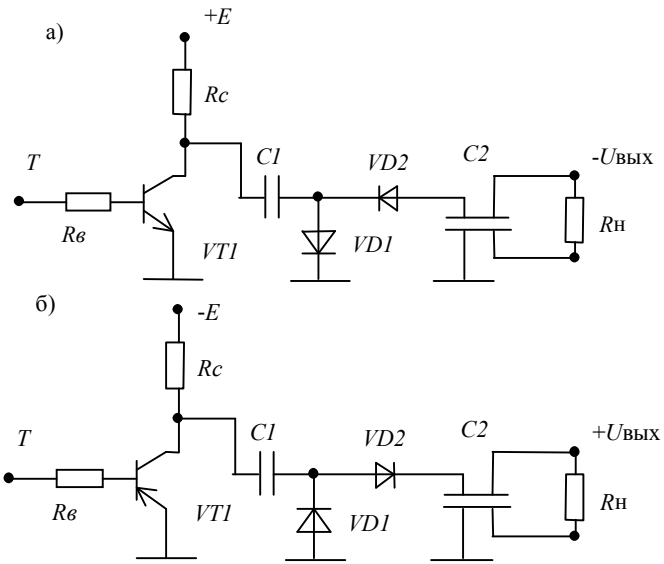



Рисунок 3.3 – Схемы конденсаторных декодеров

На вход *Т* поступает кодированный импульсный сигнал. При отсутствии импульса транзистор закрыт, и конденсатор *С1* заряжается через диод *VD1*. Во время импульса, когда транзистор открыт, конденсатор *С1* разряжается через диод *VD2* и нагрузку $R_н$. В промежутке между импульсами конденсатор *С2*, включенный параллельно нагрузке, поддерживает напряжение на выходе. Оно возрастает во время импульсов и убывает по экспоненте во время паузы. При оптимальных значениях параметров схемы напряжение на выходе близко к постоянному.

Схемы имеют единственный опасный набор неисправностей: пробой *С1* и *VD2* и обрыв *VD1*. В этом случае на выходе декодера появляется постоянное напряжение. Вероятность такого события $\lambda_{оп} < 1 \cdot 10^{-15}$ 1/ч. Кроме того, появившееся в результате отказа напряжение будет иметь обратную полярность. Если использовать чувствительную к полярности нагрузку (например, поляризованное реле), то схема не будет иметь опасных отказов.

### 3.3 Импульсные схемы с внешним тактированием

В импульсных схемах с внешним тактированием обнаружение неисправностей и автоматический переход в защитное состояние достигается благодаря импульсной работе схемы, несмотря на неизменное значение потенциалов на ее входах. Схема переключается непрерывно от импульсного сигнала, поступающего на специальный вход. Выходной сигнал получается закодированным, но для последующей обработки этот сигнал декодируется.

Примером импульсной схемы с внешним тактированием может служить схема безопасной коллекторно-базовой логики, представленная на рисунке 3.4.

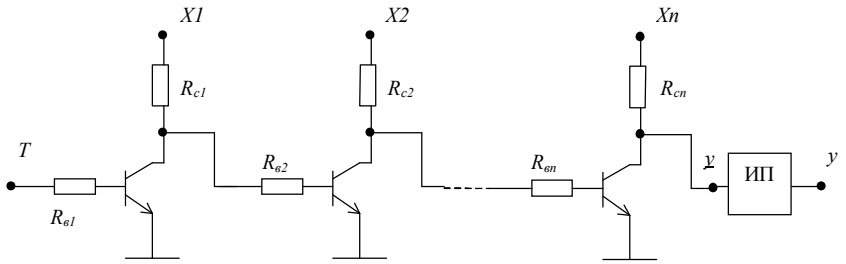



Рисунок 3.4 – Схема безопасной коллекторно-базовой логики

Логические переменные *X1, X2, ..., Xn* поступают в виде постоянного напряжения «0» или «1». На вход *T* подается импульсный сигнал. При наличии на всех логических входах высокого уровня на выходе получается импульсный сигнал той же (при четном числе каскадов) или обратной фазы (при нечетном числе каскадов), т. е. элемент реализует функцию «И» (рисунок 3.5).

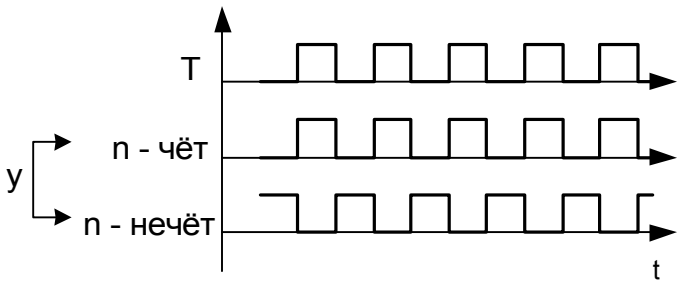



Рисунок 3.5 – Диаграмма работы схемы при различном числе каскадов

Исследования показали, что любая одиночная неисправность контролируется [1]. Двукратная – одновременное повреждение (обрыв) $R_c$ и эмиттерной цепи транзистора, где $X = 0$, – приведет к появлению на выходе инвертированного сигнала. Однако если в качестве импульсного преобразователя использовать конденсаторный декодер (см. рисунок 3.3), а скважность импульсов очень низка (К ≈ 0,1), то с поворотом фазы выходной сигнал резко уменьшается. Таким образом, опасной является только четырехкратная неисправность, т. к. может произойти двойная инверсия сигнала, которая не обнаруживается.

Пример элемента, реализующего функцию “Исключающее ИЛИ” представлен на рисунке 3.6. На вход *T* подается импульсный сигнал, который появляется на выходе при разных значениях *X1* и *X2*. При равных значениях *X1* и *X2* транзистор закрыт, и на выходе присутствует постоянное напряжение.

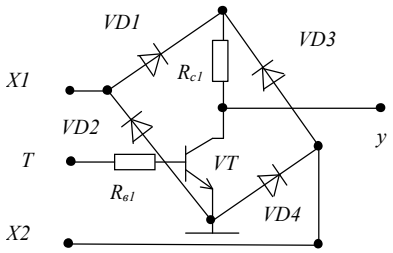



Рисунок 3.6 – Схема безопасного элемента «Исключающее ИЛИ»

Включив эти элементы последовательно, можно составить контрольный канал для сравнения результатов работы вычислительных каналов при многоканальной обработке информации.

Возможна реализация безопасных логических элементов с использованием оптронов, которые применяют для каскадной связи вместо транзисторов. На рисунке 3.7 представлена схема логического элемента «И».

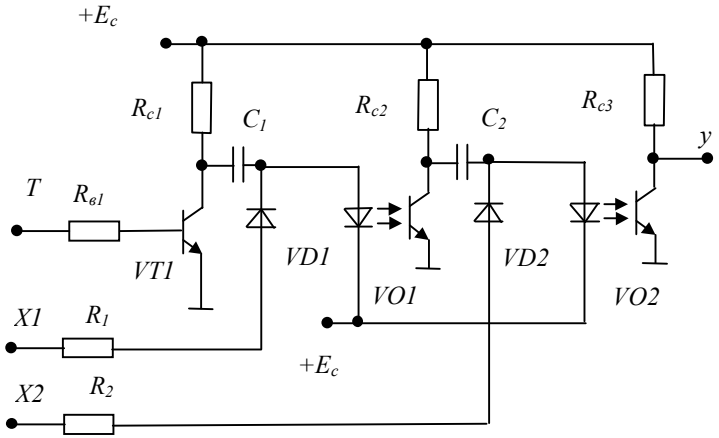


Рисунок 3.7 – Схема безопасного элемента «И»

На входы *X1* и *X2* поступают некодированные логические переменные, а на вход *T* – тактовые импульсы. В течение импульса транзисторы открываются, и конденсаторы заряжаются высоким потенциалом логических переменных. Во время паузы между импульсами конденсаторы разряжаются через светодиоды. Выходной импульсный сигнал появляется только при наличии высокого уровня на обоих информационных входах схемы. Считая невозможным пробой между светоисточником и фотоприемником оптрона, можно доказать, что любая неисправность приводит к появлению постоянного сигнала на выходе.

Если реализовать обратное включение оптронов и диодов, то схема будет вычислять логическую функцию «НЕ» (рисунок 3.8). В этом случае конденсатор будет заряжаться обратным потенциалом и разряжаться через светодиод следующего оптрона только при нулевом потенциале на входе *X*.

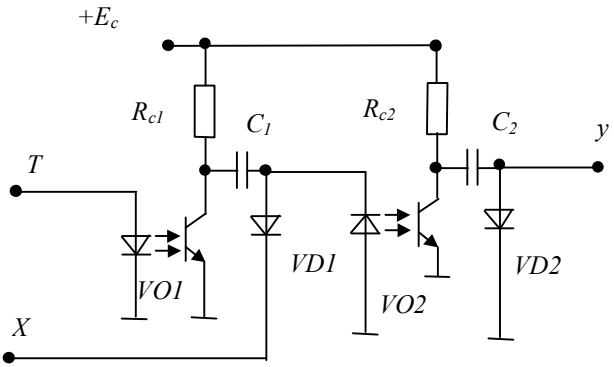



Рисунок 3.8 – Схема безопасного элемента «НЕ»

Входы и выходы в рассмотренных схемах не защищены по отношению к неисправностям типа «постоянная 1» и «постоянный 0». Контроль входов возможен только при использовании кодированных логических переменных. Однако импульсные схемы с внешним тактированием можно использовать и при однофазном или парафазном импульсном кодировании логических переменных.

В схеме элемента «И» (см. рисунок 3.7) кодированные переменные поступают на входы *T* и *X1*. В течение импульса транзистор открыт, и конденсатор заряжается. Во время паузы транзистор закрыт, и конденсатор разряжается через светодиод. Нулевой потенциал на входе *X1* не мешает работе, т. к. диод *VD1* направляет разряд только через оптрон. Если сигналы обеих переменных противофазны, то конденсатор не заряжается. Обе его пластины

изменяют потенциалы одновременно, и сигнал на выходе отсутствует. Если же требуется выдавать выходной сигнал при противофазных сигналах, то используют схему элемента «НЕ» (см. рисунок 3.8).

Логический элемент с кодированными переменными не нуждается в декодере. Декодирование происходит на выходе устройства после окончания обработки информации. Это позволяет значительно упростить схему и защитить ее от постоянных логических неисправностей на входах и выходах.

На основе коллекторно-базовой логики и оптронов созданы гибридные интегральные схемы *ELES* [1]. В корпуса размером 20×20×5 мм встроены три функциональных модуля, с помощью которых можно решать различные логические задачи.

### 3.4 Автогенераторные логические элементы

Автогенераторные логические элементы имеют следующие особенности: выходной сигнал гармонической, прямоугольной или другой формы зарождается в самой схеме; элементы работают с некодированными (устойчивыми) сигналами логических переменных; входные сигналы не обрабатываются логически, они лишь создают условия для генерации выходного сигнала.

Рассмотрим принципы построения автогенераторных логических элементов на примере логической системы *Logisafe*. В основу этой логической системы поставлен автогенератор с обратной трансформаторной связью (рисунок 3.9), коллекторная и базовая цепи которого питаются логическими переменными. Сигнал образуется в схеме, частота и форма сигнала определяется параметрами схемы. Параметры схемы рассчитывают так, чтобы условия генерации выполнялись для напряжений коллекторной и базовой цепи выше порога логической единицы. В данном случае значением логической единицы будет служить отрицательное напряжение.

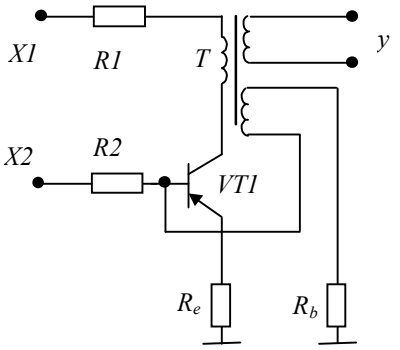



Рисунок 3.9 – Схема элемента *Logisafe*

Генератор не возбудится, каким бы большим ни было напряжение коллекторной цепи, если нет сигнала выше некоторого порога (логической «1») по базовой цепи, и наоборот. Логические переменные поступают некодированными. Так как постоянный ток не трансформируется, то значение логической перемен-

ной не может появиться на выходе схемы при любых неисправностях, за исключением короткого замыкания «вход-выход». Между сигналами на входе и выходе имеется лишь логическая связь.

Каждая логическая операция сопровождается преобразованием постоянного напряжения в гармонический сигнал автогенератора и обратно. Это реализуется по четырем последовательно связанным звеньям (трансформаторный вход для повышения напряжения и гальванического разделения каскадов, двухполупериодный выпрямитель со сглаживающим конденсатором, генератор, транзисторный усилитель) (рисунок 3.10).

Схема, представленная на рисунке 3.10, выполняет логическую операцию «И». Если необходима реализация операции «ИЛИ», оба двухполупериодных выпрямителя, на которые поступают входные сигналы *X1* и *X2*, подключаются параллельно. В схеме логического элемента «НЕ» коллекторная цепь питается от постоянного источника, а в цепь обратной трансформаторной связи подключается выпрямленный сигнал базовой цепи, куда поступает логическая переменная.

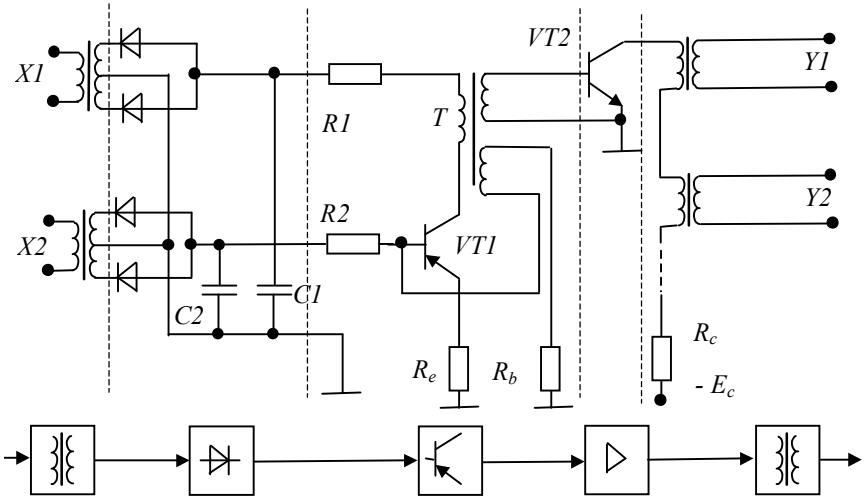



Рисунок 3.10 – Схема логического элемента «И»

Опасной является двукратная неисправность: обрыв в цепи конденсатора *C1* и пробой цепи «эмиттер-коллектор» *VT1*, когда пульсации питающего напряжения коллекторной цепи в результате прямой трансформации появляются на выходе. Избежать такой неисправности можно введением четырехвыводного конденсатора во входную цепь. Благодаря транзисторному

усилителю VT2 можно подключать большое число входов следующих каскадов.

Систему *Logisafe* выпускает фирма *Telefunken* [1]. Все модули системы оформлены в виде гибридных ИС размерами 44×11×18 и 44×21×18 мм. В каждой ИС содержится до трех одинаковых модулей.

### 3.5 Самопроверяемые логические элементы

Самопроверяемые элементы обладают способностью обнаружения отказов в процессе нормального функционирования. Отказы обнаруживаются по значениям выходных сигналов без дополнительной подачи на входы элементов специальных проверочных тестов или других способов проверки. Самопроверяемые элементы позволяют строить сложные самопроверяемые схемы, в которых любая неисправность распространяется от точки своего возникновения к выходам схемы. Если на выходе схемы стоит контрольный элемент, то таким образом можно легко контролировать исправность всей схемы.

Рассмотрим самопроверяемые элементы, работающие в парафазной логике. В этом случае значение логической «1» кодируется как «10», а значение «0» – как «01». Коды «00» и «11» являются защитными.

Схемы парафазных элементов представлены на рисунке 3.11.

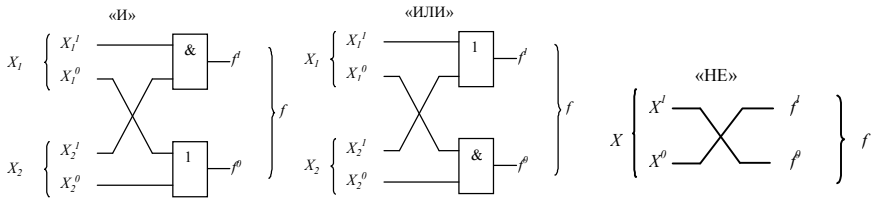



Рисунок 3.11 – Схемы парафазных элементов

В таблице 3.1 приведены значения выходов парафазного элемента «И» в исправном состоянии и при наличии константных неисправностей.

*Таблица 3.1*

| Входы | | Выход | Неисправности | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | $f$ | $x_1^1=0$ | $x_1^1=1$ | $x_1^0=0$ | $x_1^0=1$ | $x_2^1=0$ | $x_2^1=1$ | $x_2^0=0$ | $x_2^0=1$ | $f^1=0$ | $f^1=1$ | $f^0=0$ | $f^0=1$ |
| 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 11 | 00 | 01 |
| 01 | 10 | 01 | 01 | 11 | 00 | 01 | 01 | 01 | 01 | 01 | 01 | 11 | 00 | 01 |
| 10 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 11 | 00 | 01 | 01 | 11 | 00 | 01 |
| 10 | 10 | 10 | 00 | 10 | 10 | 11 | 00 | 10 | 10 | 11 | 00 | 10 | 10 | 11 |

Из таблицы 3.1 следует, что для каждой одиночной неисправности выходной сигнал элемента может быть правильным или принимать защитное значение при одном из наборов проверяющего теста {01 10, 10 01, 10 10}. Аналогично функционирует элемент при нарушении парафазности на его входах. Такими же свойствами обладают элементы «ИЛИ» и «НЕ».

Любая комбинационная схема в результате замены элементов «И», «ИЛИ», «НЕ» их парафазными реализациями преобразуется в самопроверяемую схему, обладающую следующими свойствами:

- любая комбинация константных неисправностей приводит к нарушению парафазности на выходе хотя бы при одном наборе данных проверяющего теста;
- нарушение парафазности хотя бы на одном из входов схемы приводит к нарушению парафазности на выходе хотя бы при одном наборе данных проверяющего теста.

В качестве элементов памяти в самопроверяемых схемах используют самопроверяемые парафазные триггеры. Рассмотренный в п. 2.2.2 парафазный *T*-триггер (см. рисунок 2.13) используется в качестве элементарной ячейки памяти. На его основе можно реализовать триггер любого типа [1].

Самопроверяемые логические элементы и триггеры образуют функционально полный набор элементов для реализации самопроверяемых схем без применения избыточного кодирования.

Таким образом, нами рассмотрены основные принципы построения безопасных логических элементов и типовые схемные решения. Безопасные логические элементы используются на аппаратном уровне защиты для создания контрольных элементов МИУС, непосредственно отвечающих за безопасность. В следующей главе рассмотрим принципы построения безопасных схем включения исполнительных реле и организацию безопасного ввода информации в МИУС.

# 4 ОРГАНИЗАЦИЯ БЕЗОПАСНОГО ИНТЕРФЕЙСА С ОБЪЕКТОМ УПРАВЛЕНИЯ

Важной проблемой при построении безопасных систем является организация сопряжения микроэлектронной аппаратуры железнодорожной автоматики и телемеханики с исполнительными объектами (ИО). Известны два основных подхода к организации интерфейса в МИУС:

- жесткая унификация и стандартизация входных и выходных параметров элементов МИУС;
- использование специализированных функциональных блоков с адаптивными характеристиками по входам-выходам.

В настоящее время производится и эксплуатируется много различных исполнительных объектов железнодорожной автоматики и телемеханики с разнообразными характеристиками по входам-выходам. В цепях железнодорожной автоматики и телемеханики, к которым не предъявляются требования безопасности, как правило, применяют стандартные УСО, выпускаемые промышленностью в составе управляющих ЭВМ и контроллеров. При разработке микроэлектронных систем железнодорожной автоматики, к которым предъявляются специфичные для ответственных систем требования безопасности, чаще используются специализированные устройства сопряжения.

## 4.1 Требования к специализированным УСО

К УСО СЖАТ предъявляются следующие основные требования [1, 2]:

- обеспечение минимально допустимой вероятности возникновения ложного сигнала включения ИО на выходе УСО при любом отказе его элементов;
- выполнение временного и энергетического согласования электронных схем и ИО;
- сохранение временных и энергетических параметров УСО в заданных пределах в течение всего срока эксплуатации;
- обеспечение минимально допустимой чувствительности к электромагнитным помехам и влияниям;

• высокая технологичность производства в сочетании с низкой стоимостью.

Схемные решения устройств сопряжения с объектами (УСО) СЖАТ не должны иметь опасных отказов, т. е. с определенной вероятностью должны исключать ложное включение ИО на выходе УСО при любом отказе его элементов. Обычно учитываются следующие отказы:

• короткое замыкание;

• обрыв элементов или соединений;

• трансформация одного типа полупроводникового элемента в другой;

• самовозбуждение электронных схем;

• кратковременное или длительное отключение источника питания;

• повреждение источника питания, при котором на его шинах появляется значительная переменная составляющая;

• изменение параметров элементов или режимов их работы в установленных пределах;

• появление двух или более отказов элементов или соединений, не выявленных за время нахождения схемы в статическом состоянии.

Для исключения накопления отказов УСО, как правило, строятся по принципу обеспечения динамической работы всех элементов, что позволяет диагностировать электронные элементы путем периодического переключения их из состояния логической «1» в логический «0» и обратно.

### 4.2 Устройства включения исполнительных реле

Условно УСО можно разделить на две части: устройства вывода управляющей информации и устройства ввода контрольной информации о состоянии исполнительных объектов. В зависимости от используемой элементной базы УСО можно разделить на релейные и электронные (бесконтактные).

Наиболее часто безопасные УСО выполняют в виде функциональных преобразователей с несимметричным отказом, у которых при появлении неисправности искажаются передаточные функции. Возникающие в этом случае выходные сигналы неисправного УСО не должны приводить к ложному включению исполнительного объекта. Широко применяют функциональные преобразователи с использованием специальных реле I класса надежности, контакты которых коммутируют рабочие цепи исполнительного объекта. Такие УСО называют устройствами включения исполнительных реле (УВИР).

Преимуществами такого решения является то, что реле имеют высокую устойчивость к электромагнитным помехам и перенапряжениям, являются элементами идеальной гальванической развязки с несимметричным отказом.


5

Недостатки реле состоят в ограниченном ресурсе и потребности в профилактическом обслуживании релейно-контактных схем, а также специфичности производства релейных приборов.

Рассмотрим основные принципы построения безопасных устройств включения исполнительных реле. Наиболее распространена схема УВИР, представленная на рисунке 4.1. Безопасное поведение этой схемы при отказах обеспечивается за счет двойного преобразования входных импульсных сигналов – дифференцирования с помощью трансформатора и интегрирования с помощью диода и конденсатора. При нарушении любого из этих двух законов преобразования сигналов на выходе схемы либо отсутствует напряжение, либо оно меньше напряжения выключения, поэтому реле ИР отпустит свой якорь.

Для включения реле, т. е. для достижения напряжения срабатывания, необходимо поступление на вход схемы серии импульсов. Кратковременные случайные сбои в работе СЖАТ не приводят к ложному выключению или включению ИР из-за инерционности заряда и разряда конденсатора. Во включенном состоянии реле находится до тех пор, пока поступают импульсные сигналы.

Недостатком УВИР с трансформаторной гальванической развязкой является то, что импульсные трансформаторы – нетехнологичные элементы. Поэтому в последнее время разрабатываются УВИР с конденсаторной гальванической развязкой (рисунок 4.2).

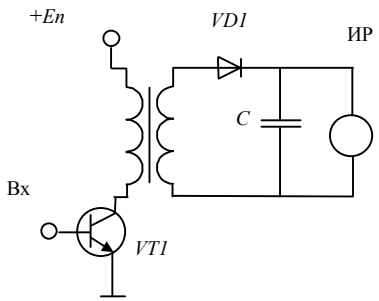



Рисунок 4.1 – УВИР с трансформаторной гальванической развязкой

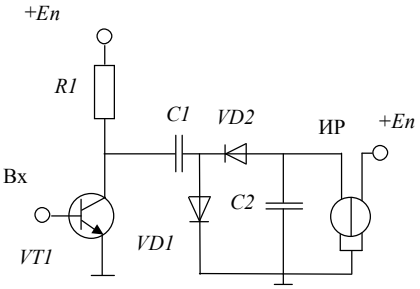



Рисунок 4.2 – УВИР с конденсаторной гальванической развязкой

Рассмотренные УВИР имеют один вход и могут использоваться в микроэлектронных системах, достоверность выходных сигналов которых контролируется специальными средствами. В дублированных системах могут использоваться УВИР, принцип работы которых основан на преобразовании

импульсных сигналов малой амплитуды в рабочее напряжение ИР с помощью выпрямителей с умножением напряжения (рисунок 4.3).

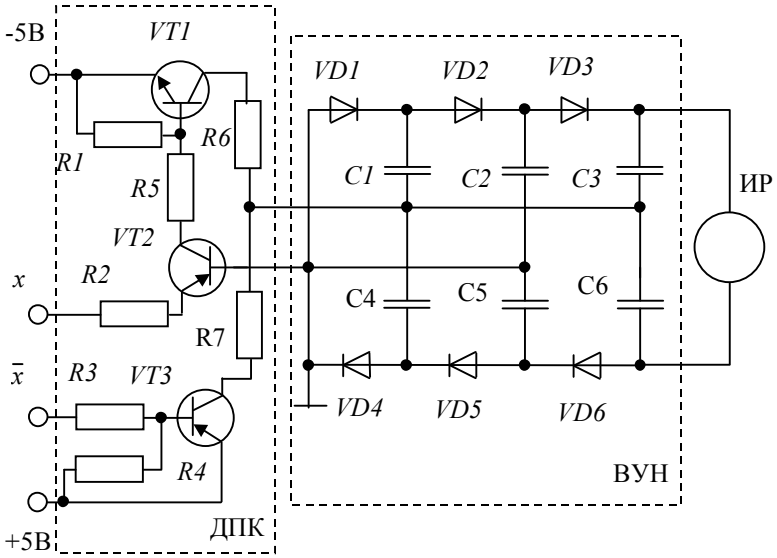



Рисунок 4.3 – УВИР на основе выпрямителя с умножением напряжения

В схеме УВИР (см. рисунок 4.3) входные сигналы в виде последовательности импульсов поступают на прямой и инверсный входы двухполюсного ключа (ДПК). При парафазности сигналов, поступающих от разных вычислительных каналов, на входе выпрямителя с умножением напряжения (ВУН) появляется переменное напряжение прямоугольной формы с амплитудой, меньшей напряжения отпускания реле ИР. ВУН выпрямляет и умножает исходное напряжение до уровня, необходимого для срабатывания ИР при поступлении нескольких импульсов.

Повреждение любого элемента УВИР ведет к прекращению умножения напряжения или снижению выходного напряжения ВУН, что исключает возможность ложного включения реле ИР. Кроме того, УВИР контролирует правильность работы двух каналов (парафазность выходных сигналов), выполняя роль выходных компараторов.

В микроэлектронных системах, выполненных по мажоритарной структуре «2 из 3», можно использовать УВИР, представленное на рисунке 4.4. При синхронном поступлении импульсных сигналов на входы 1, 2, 3 происходит заряд конденсаторов $C_1$, $C_2$, $C_3$ в течение времени действия входных импульсов. Во время паузы они разряжаются на светодиоды оптопар $VO_1$ и $VO_2$ че-

рез резисторы $R_1$, $R_2$. При этом напряжение, воздействующее на них, равно сумме напряжений на конденсаторе и источнике питания. В результате этого фототранзисторы оптопар переключаются и формируют импульсы, поступающие на вход преобразователя полярности (элементы $C_4$, $C_5$, $VD_6$, $VD_7$). Поляризованное реле ИР притягивает якорь.

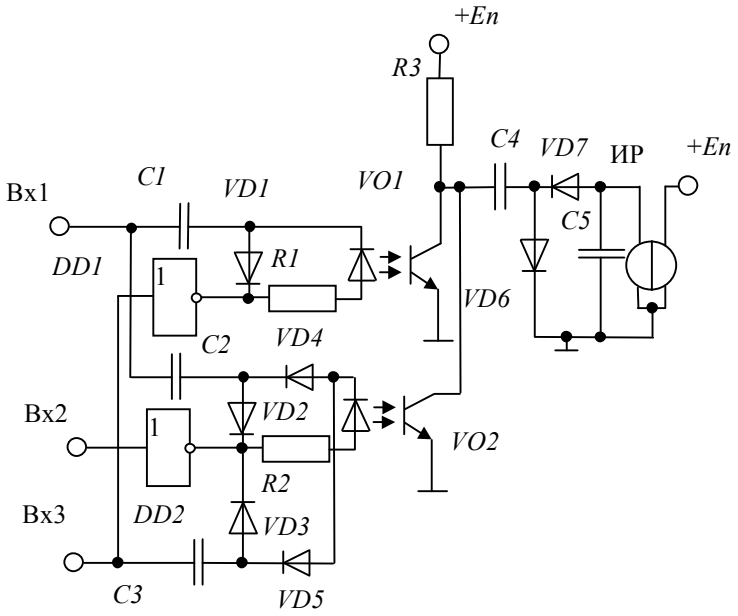



Рисунок 4.4 – Мажоритарное УВИР

При отсутствии импульсов на двух входах из трех на светодиоды воздействует только напряжение заряда конденсаторов приблизительно в два раза меньшей величины, чем в случае синхронного поступления импульсных сигналов на все входы. В результате оптроны не переключаются, и ИР отпустит свой якорь. В данной схеме для обеспечения ее безопасного функционирования используются функции дифференцирования, удвоения напряжения и гальванической развязки.

### 4.3 Безопасный ввод информации

Для обеспечения необходимой достоверности контрольной информации о состоянии исполнительных объектов в безопасных системах используются различные виды избыточного кодирования последовательного или парал-

лельного вида. Рассмотрим примеры построения устройств безопасного ввода информации о состоянии релейных датчиков (рисунок 4.5 *а, б*).

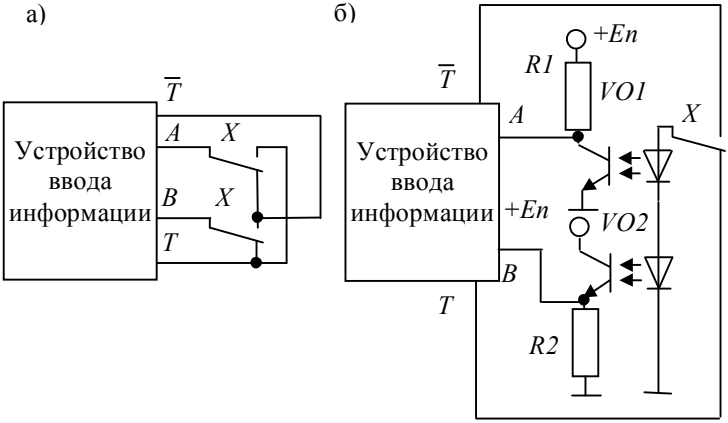



Рисунок 4.5 – Устройства безопасного ввода информации

На выходах устройства ввода информации $T$ и $\overline{T}$ генерируются последовательности парафазных импульсов. Достоверность информации подтверждается парафазностью импульсных последовательностей, поступающих на входы $A$ и $B$. Значение переменной $X$ определяется следующим образом. Если на вход $A$ приходит сигнал $T$, а на вход $B$ – сигнал $T$, то контакт реле замкнут, в противоположном случае – контакт разомкнут.

При неисправности нарушается парафазность или импульсный характер сигналов на входах $A$ и $B$, что фиксируется с помощью программных или аппаратных средств контроля устройства ввода информации.

Таким образом, можно сформулировать основные принципы построения устройств сопряжения объектов, отвечающих требованиям безопасности и выполненных с использованием реле 1-го класса надежности:

- обеспечение непрерывного контроля исправности электронных элементов путем периодического изменения их состояния (принцип динамического контроля);
- статистическая обработка сигналов включения исполнительных реле;
- гальваническая развязка входных и выходных цепей;
- частотная или амплитудная защита схемы от неисправностей источника питания;
- отсутствие обратных связей, приводящих к самовозбуждению схем;
- амплитудная, полярная или частотная защита УСО от ложного включения исполнительных реле.

# 5 ПРОГРАММНЫЕ МЕТОДЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

Требование высокой надежности является первостепенным при проектировании систем железнодорожной автоматики. При построении таких систем на базе программно-управляемой аппаратуры их специализация под конкретные технологические задачи производится программным способом. Программные средства в этом случае являются определяющими в реализации системой требуемых функций.

Создание безопасных систем, логика функционирования которых отражена в виде программы, охватывает два аспекта:

- безопасность собственно программного обеспечения (ПО);
- безопасность системы, гарантируемая средствами программного обеспечения.

Рассмотрим структуру программного обеспечения микропроцессорных систем железнодорожной автоматики на примере ПО микропроцессорной централизации и программные методы обеспечения безопасности.

## 5.1 Структура программного обеспечения микропроцессорной централизации

Программное обеспечение микропроцессорной централизации состоит из двух больших блоков – системного и прикладного. Ядром системного ПО является специализированная операционная система (ОС) реального времени. Функции операционной системы не зависят от непосредственных задач системы централизации и заключаются в выделении ресурсов процессора для решения различных функциональных задач. Операционная система определяет:

- очередность выполнения вычислительных процессов;
- обслуживание и управление датчиками времени;
- назначение прерываний;
- управление выдачей синхронизирующих сигналов.

Таким образом, ОС полностью отстраняет прикладные программы от непосредственного управления вычислительным процессом.

Операционную систему дополняют пакеты программ, ориентированные на техническое обеспечение системы. Функциями этих пакетов программ являются:

- повторный запуск системы, включая проверки и синхронизацию;
- актуализация ЭВМ (подключение восстановленного канала без перерыва в работе системы);
- обслуживание прерываний;
- проверка ЭВМ в реальном масштабе времени;
- управление интерфейсами отдельных блоков (выдача команд напольному оборудованию, управление последовательными интерфейсами, сопряжением с системой шин);
- специальные функции времени (управление таймером, тактирование системы).

Прикладное ПО реализует функции системы централизации. Оно состоит из модели станции и пакетов прикладных программ, реализующих типовые функции.

Модель станции состоит из двух частей: модели конфигурации и модели состояния. Модель конфигурации является неизменяемой частью и содержит информацию о конфигурации станции, размещении напольного оборудования, внешних системах (автоблокировке (АБ), диспетчерской централизации (ДЦ), автоматической локомотивной сигнализации (АЛС) и других). Модель состояния содержит оперативную информацию о состоянии напольного оборудования, поездном положении, текущем состоянии системы, отказах и сбоях технических средств и т.п.

Пакеты прикладных программ можно условно разделить на несколько типовых программ, решающих основные функции централизации и функции взаимодействия с оперативным персоналом и напольным оборудованием.

Пакет «Ответственные функции» решает логические задачи обеспечения безопасности поездных и маневровых маршрутов. Здесь осуществляется проверка допустимости маршрутов, их приготовление, контроль, замыкание и размыкание.

Пакет «Управление и индикация» функционально адаптирует к соответствующим требованиям пользователя клавиатуру, табло, дисплей и печатающее устройство для протоколирования эксплуатационного процесса и регистрации нарушений.

Пакет «Подключение напольных устройств» имеет модульную структуру. Он содержит программы для выдачи команд: установки требуемых сигнальных показаний; перевод стрелок; контроля и извещения об исправных состояниях светофоров, стрелок, рельсовых цепей; распознавания сбоев или отказов в работе напольного оборудования и т.п.

Пакет «Сопряжение с внешними системами» содержит программы для организации связи с устройствами АБ, ДЦ, АЛС, индикации номеров поездов, телеуправления и телесигнализации.

Как видно из структуры ПО микропроцессорной централизации, не все программы в одинаковой степени отвечают за безопасность. Поэтому и методы обеспечения безопасности для каждого программного модуля могут быть различны.

## 5.2 Основные принципы обеспечения безопасности программного обеспечения

Одной из причин нарушения работоспособности программных средств является отклонение исходного текста программы от формализованного эталона и требований заказчика. Ошибки такого рода в практике программирования получили название ошибок программирования. Ошибки второго рода определяются прежде всего отказами и сбоями аппаратных средств. Из-за высокой степени интеграции аппаратно-программных средств большинство отказов и сбоев аппаратуры (регистров, ячеек памяти, информационных магистралей) приводят к искажению команд программы и используемых данных.

В процессе эксплуатации надежность ПО выражается через свойства ее устойчивости и безопасности. Эти свойства, так же как и аналогичные свойства аппаратных средств, обеспечиваются введением избыточности, которая может быть структурной, информационной и временной. Дополнительные программные средства должны обеспечивать последовательное решение задач обнаружения искажения алгоритмов функционирования, ограничения последствия этого искажения в пределах некоторого участка программы (программного модуля) и восстановления правильного результата. В безопасных управляющих программах вместо восстановления возможен перевод системы в защитное состояние.

*Временная избыточность* состоит в выделении специальных интервалов времени для организации процедур контроля и восстановления. При этом функциональные задачи не решаются, поэтому производительность вычислительной системы уменьшается. Широко применяют предстартовый функциональный контроль с использованием контролирующих и диагностических тестов или решение контрольных задач. Сохранность программ проверяется суммированием кодов программы и сравнением результата с контрольной суммой. В системах управления, работающих в реальном масштабе времени, временная избыточность используется, если существуют технологические перерывы в работе систем, во время которых осуществляется тестирование или повторный счет.

*Информационная избыточность* заключается в резервировании информационных массивов и в применении корректирующих кодов для представления информации. В случае разрушения основного информационного массива программа обращается к резервному, который используется до полного восстановления основного массива. Эффективным способом обеспечения устойчивости является организация хранения дополнительной информации о текущем состоянии программы в контрольных точках. Это состояние сохраняется для восстановления вычислительного процесса с ближайшей контрольной точки от места возникновения ошибки.

*Структурная избыточность* состоит в использовании методов *n*-вариантного и самопроверяемого программирования. Варианты одной и той же программы могут быть одинаковыми или различаться методами решения задачи или способами программной реализации одного и того же метода. Целесообразно также, чтобы разные варианты программ были написаны различными бригадами программистов. При исполнении программы результат вычислений выбирают голосованием. Последовательная реализация программ требует больших затрат времени, поэтому *n*-вариантное программирование используют обычно в многопроцессорных вычислительных системах. В безопасных системах часто применяют двухвариантное программирование с контролем совпадения результатов. На применение в безопасных системах ориентированы также самопроверяемые программы.

Таким образом, все принципы и методы обеспечения надежности и безопасности ПО в соответствии с их целью можно разбить на три группы: предупреждение ошибок, обнаружение ошибок, исправление ошибок и обеспечение устойчивости к ошибкам. К первой группе относятся принципы и методы, позволяющие минимизировать или вообще исключить ошибки программирования. Методы второй группы направлены на выявление ошибок в программном обеспечении. К третьей группе относятся методы, предназначенные для исправления ошибок или их последствий, т.е обеспечивающие устойчивость к ошибкам и позволяющие программному обеспечению правильно функционировать при наличии ошибок в программном обеспечении или отказах и сбоях аппаратных средств.

Предупреждение ошибок – наиболее эффективный путь обеспечения надежности, так как ошибку легче предупредить, чем потом обнаружить и исправить. Методы предупреждения ошибок в основном применяются на этапе проектирования. Для предупреждения ошибок используют рациональную организацию труда разработчиков ПО, методы структурного проектирования и автоматизацию проектирования [7]. Эти методы позволяют систематизировать и упростить процессы проектирования ПО, сделать реальной задачу написания корректных программ с малым количеством ошибок.

Рассмотрим остальные методы обеспечения надежности и безопасности ПО более подробно.

## 5.3 Обнаружение искажения вычислительного процесса

### 5.3.1 Обнаружение программных ошибок

Меры по обнаружению ошибок можно разбить на две группы: пассивные методы обнаружения ошибок в процессе нормальной работы программного обеспечения и активные методы проверки программного обеспечения в поисках ошибок.

Пассивные методы могут быть приняты на различных структурных уровнях программного обеспечения при передаче управления от одного программного модуля другому. Пассивные методы базируются на следующих принципах:

- взаимном недоверии, т. е. каждый модуль должен предполагать, что все другие модули и входная информация содержат ошибки;
- немедленном обнаружении, т. е. ошибки должны обнаруживаться как можно раньше, поэтому все проверки необходимо производить сразу после получения данных;
- избыточности, т. е. если во входных данных имеется избыточность, она должна использоваться для проверки, если избыточности нет – необходимо ее ввести (например, контрольную сумму);
- регистрации ошибок;
- блокировке системы в защитном состоянии.

Пассивные методы могут обнаружить ошибку только в том случае, если она исказила данные заранее предусмотренным в средствах проверки способом. Однако это далеко не всегда возможно. Поэтому в высоко надежных системах пассивные методы применяются вместе с активными методами обнаружения ошибок.

Основным активным методом обнаружения ошибок программирования является их тестирование, в основе которого лежит тот факт, что программа любой сложности при строго фиксированных исходных данных и абсолютно надежной аппаратуре исполняется однозначно определенному алгоритму. Исполнение всех маршрутов программ является сложной комбинаторной задачей, объем которой определяется произведением:

$$N_t = N_{vh}\, N_s\,, \tag{5.1}$$

где $N_t$ – число тестов;

$N_{vh}$ – число всех возможных входных ситуаций;

$N_s$ – число состояний вычислений.

Для уменьшения времени тестирования необходимо использовать методы упорядочивания и систематизации процесса, а также методы тестирования по различным стратегиям и параметрам [1, 7, 8].

Целью тестирования является обнаружение отказов аппаратных средств для предотвращения их влияния на выполнение основного алгоритма. В структурированной программе выделяют тестирование модулей, сопряжений между модулями, внешних функций и комплексное тестирование.

На уровне модулей проверяют логику программы. Контроль сопряжений обнаруживает ошибки в межмодульном интерфейсе. Тестирование внешних функций определяет соответствие внешних спецификаций и функций программы. Комплексное тестирование является завершающим этапом проверки системы.

При составлении тестов используются два подхода: функциональный и структурный. В первом случае внутренняя структура программы не учитывается. Тесты составляются на основании функциональных спецификаций. Критериями оценки полноты функционального теста являются проверки:

1) всех классов входных данных, когда тест должен содержать по одному представителю из каждого класса;

2) всех классов выходных данных, когда при исполнении тестовых примеров должно быть получено по одному представителю из каждого класса;

3) всех функций, когда каждая реализуемая программой функция должна быть проверена хотя бы 1 раз;

4) всех ограничений и правил и т. п.

Структурные тесты строятся с использованием информации о внутренней структуре программы. Критериями оценки полноты тестирования являются проверки:

1) каждой команды – не менее 1 раза;

2) каждой ветви программы – не менее 1 раза;

3) каждого пути (последовательности команд) программы – не менее 1 раза.

Если программа содержит циклы, то проводят тестирование простых, ациклических путей, а число итераций ограничивают некоторой константой.

Процесс контроля методом тестирования предусматривает поочередное выполнение тестовых и программных процедур. При этом их взаимное чередование может быть произвольным и определяться объемом теста. В микропроцессорных системах нашло широкое применение тестирование технических средств методом раскрутки, т. е. сначала тестируется ядро системы (при этом учитывается способность микропроцессора к самотестированию), а затем микропроцессор принимает участие в процессе тестирования окружения.

Преимуществами тестирования являются увеличение глубины контроля и упрощение процесса создания программ технологических алгоритмов, т. к. процесс создания тестового программного обеспечения может рассматриваться как процесс независимый, ограниченный только параметрами управляемого технологического процесса. Недостатки тестирования: не обнаруживаются аппаратные сбои, во время тестирования состояние объекта управления не контролируется. Кроме того, даже для несложных программ тестирование не может доказать отсутствие в них ошибок, а может только обнаружить некоторую их часть. Поэтому тестирование используется как дополнительное средство контроля совместно с другими методами.

### 5.3.2 Обнаружение отказов и сбоев аппаратных средств

Степень безопасности при отказах и сбоях аппаратных средств определяется возможностью средств контроля адекватно оценить состояние технических средств в заданный промежуток времени. Ввиду высокой сложности микропроцессорных систем в качестве средств контроля необходимо применять как аппаратные, так и программные средства. Использование ПО как средства контроля правильности функционирования имеет свою специфику, которую надо учитывать при создании безопасных программно-управляемых систем.

Последовательный характер обработки команд программы приводит к необходимости чередования процедур основной программы с контрольными. В этом случае увеличивается период времени между возникновением неисправности и ее обнаружением. Аппаратная зависимость программных процедур контроля, т. е. использование для реализации основной программы и программ контроля одних и тех же технических средств, ставит под сомнение результаты работы контрольной программной процедуры, т. к. ошибки, искажающие основной алгоритм, искажают и алгоритм контрольной программы.

Поэтому программные методы контроля чаще всего дополняют аппаратные средства контроля и ориентированы на проверку результатов вычислений. Наиболее часто используемыми методами программного обнаружения отказов и сбоев аппаратных средств являются создание самопроверяемых программ и *n*-версионное программирование. Кроме того, *n*-версионное программирование позволяет обнаруживать и некоторые программные ошибки.

### 5.3.3 Самопроверяемые программы

Программа называется самопроверяемой, если для любого входа, удовлетворяющего входной спецификации, программа выдает правильный выход-

ной результат или вырабатывает предупреждающее сообщение, информирующее пользователя, что выход может быть неправильным.

Целью создания самопроверяемых программ является получение программ, удовлетворяющих свойствам защищенности и самотестируемости относительно определенного класса неисправностей аппаратных средств.

Управляющая программа называется защищенной, если при возникновении любой ошибки заданного класса на любой рабочей последовательности входных данных, выходные данные вычисляются правильно или являются защитными.

Управляющая программа называется самотестируемой, если для каждой ошибки заданного класса существует хотя бы одна рабочая последовательность входных данных, на которой вычисляется хотя бы одно защитное значение выходных данных.

Таким образом, управляющая программа будет самопроверяемой, если она защищена от ошибок и является самотестируемой.

Защищенность от ошибок исключает неправильные воздействия со стороны программной системы на управляемые объекты. Самотестируемость исключает наличие необнаруживаемых ошибок и их накопление с течением времени. Эти два свойства являются основными требованиями к безопасным системам.

Метод построения самопроверяемых программ базируется на результатах синтеза самопроверяемых конечных автоматов. Для заданного алгоритма управления находят функциональное описание выполняющего этот алгоритм автомата. Кодирование состояний автомата производится равновесным кодом. Данное свойство позволяет контролировать состояние переменных и вычислять защитный вектор (вектор, отличный от кодового слова) в случае их искажения. При этом контроль слова состояния автомата производится программной реализацией самопроверяемой комбинационной схемы [9, 10].

Таким образом, самопроверяемая программа представляет собой последовательность программных модулей, одни из которых выполняют собственно алгоритм управления, а другие – контролируют правильность этого вычисления. Результаты вычислений каждого модуля анализируются в программных контрольных точках.

Достоверность контроля зависит от вероятности того, что искажение результата на выходе модуля обнаруживается с помощью утверждения, которому должно удовлетворять состояние расчета программы в данной контрольной точке. Несовершенство данного метода состоит в том, что если некоторые операторы контроля указывают ошибку, то система (или программист) делает вывод о неисправности программы. Но если все операторы контроля указывают истинность, то нельзя с уверенностью заключить,

что программа правильна.

Повысить достоверность контроля можно двумя способами. Первый способ состоит в уменьшении периода контроля $\tau_k$. Время $\tau_k$ равно времени исполнения программы контролируемого модуля. При уменьшении размеров модуля при $\tau_k \to 0$ улучшаются показатели надежности программы, но возрастают размеры программ. Второй способ заключается в таком выборе утверждений и в таком их размещении в программе, которые исключают (при заданных ограничениях) ложный вывод о правильности результата вычислений. Примеры реализации самопроверяемых программ, вычисляющих функции алгебры логики и автоматов, приведены в [9, 10].

Достоинством данного метода является возможность сочетания программного контроля с аппаратными средствами контроля, основанными на использовании парафазной логики.

Недостатки – увеличение кода программы; обнаружение только неисправностей, приводящих к искажению кодового вектора слова состояния автомата; охватывает только класс логических алгоритмов управления.

### 5.3.4 *N*-версионное программирование

Целью *n*-версионного программирования является обнаружение оставшихся ошибок проектирования программного обеспечения и отказов аппаратных средств с целью предотвращения опасных отказов.

*N*-версионное программирование предусматривает *n*-разовую реализацию данной программы различными способами. Эффективность данного метода определяется прежде всего степенью непохожести программных компонент (диверситетом), сводящих к минимуму появление одинаковой реакции при нарушении работы технических средств или наличии программных ошибок.

Диверситет программ может быть достигнут:

- созданием версий программы разными программистами или коллективами программистов или использованием упрощенной модели программы в качестве другой версии;
- использованием разных методов логической организации программ;
- использованием различных версий языков или разных версий компиляторов с одного и того же языка.

Результаты работы различных версий программ сравниваются. При различии результатов фиксируется ошибка. Сравнение может быть организовано аппаратно или программно. В последнем случае для корректного сравнения результатов работы вариантов программ могут использоваться самопроверяемые программы. Данный подход позволяет контролировать как состояние технических средств, так и наличие программных ошибок. К недостаткам можно отнести значительное увеличение объектного кода про-

граммы и затрат на разработку.

### 5.4 Исправление ошибок и устойчивость к ошибкам

Под исправлением ошибок в программном обеспечении понимается устранение ущерба, нанесенного ошибкой, а не исправление самой ошибки. Это связано с тем, что пытаться исправить ошибку в программном обеспечении без участия человека невозможно. Самое большее, что можно сделать в этом случае – обеспечить устойчивость ПО к ошибкам и определить безопасное поведение при отказах и сбоях технических средств для того, чтобы свести нанесенный ошибкой ущерб к минимуму.

Основными методами исправления ошибок и обеспечения устойчивости к ошибкам являются: использование отказоустойчивого кодирования информации, защищенное программирование, локализация отказов в рамках некоторой части ПО, безопасное поведение при отказах.

Использование отказоустойчивого кодирования ответственной информации, хранящейся в ячейках памяти и регистрах, позволяет при возникновении отказов технических средств не только обнаружить искажение информации, но и восстановить ее первоначальное значение.

Целью защищенного программирования является получение программ, которые обнаруживают ошибки, проявляющиеся в аномальных передачах управления, передачах данных, разрушении части объектного кода, и реагируют на них заранее определенным образом.

При реализации защищенного программирования можно выделить несколько технических приемов:

а) уменьшение разрушающего влияния ошибок на программу. Эти методы основаны:

- на детальном анализе команд конкретной вычислительной системы с целью прогнозирования поведения системы при возможной их модификации из-за возникновения неисправностей аппаратных средств;
- ограничении использования команд определенного типа;
- введении в структуру программы команд, выполняющих функции компенсаторов, пассивных с алгоритмической точки зрения, но активных с точки зрения контроля;

б) диагностическая модификация основной программы, позволяющая отслеживать фактическое выполнение последовательности команд, целостность структуры программы.

Данный метод хорошо формализуем, что позволяет автоматизировать процесс модификации исходного текста программы с учетом конкретной спецификации команд, видов программ, учитываемого класса отказов.

К недостаткам защищенного программирования можно отнести увеличение объектного кода, наличие ограничений на процесс программирования, отсутствие контроля искажения данных программы.

Локализация отказов в рамках некоторой части ПО достигается использованием при программировании следующих правил:

- никакой модуль программы не должен иметь возможности непосредственно использовать данные других модулей или операционной системы. Связь между двумя программами может быть разрешена только при условии использования четко определенного интерфейса и только в случае, когда обе программы дают на эту связь согласие;
- программы и их данные должны быть защищены от ошибок и сбоев в работе операционной системы, чтобы эти ошибки не могли привести к случайному изменению прикладных программ или их данных;
- операционная система должна защищать все прикладные программы и данные от случайного их изменения пользователем или другими программами.

Эти правила в основном распространяются на организацию взаимодействия прикладных программ между собой и с операционной системой. Использование этих правил не противоречит принципам структурного программирования, а их реализация зависит от структуры аппаратного и программного обеспечения системы.

Безопасное поведение при отказах обеспечивается использованием следующих методов:

- повторное выполнение участков программ, при выполнении которых зафиксирован отказ (сбой). Повтор может производиться либо с последней команды (функции, процедуры), либо с некоторой контрольной точки. В этом случае под контрольной точкой понимается периодически обновляемая копия состояния прикладной программы или всей системы;
- резервное копирование и восстановление искаженных участков памяти или файлов;
- динамическое изменение конфигурации системы при отказах некоторых ее подсистем;
- отключение выходов системы для исключения неверного воздействия на объект управления.

Таким образом, в микропроцессорных системах можно добиться высокой степени защищенности управляющих программ относительно программных ошибок и отказов аппаратных средств, используя методы предупреждения, обнаружения и обеспечения устойчивости к ошибкам.

# 6 ПОРЯДОК РАСЧЕТА ПОКАЗАТЕЛЕЙ БЕЗОТКАЗНОСТИ И БЕЗОПАСНОСТИ МИУС

Для микропроцессорных систем железнодорожной автоматики, кроме задачи синтеза безопасной системы, обязательной является задача анализа достигнутого при разработке уровня безопасности. Одним из основных методов анализа является расчетный метод. В данной главе рассмотрены основные методы расчета показателей безотказности и безопасности микропроцессорных систем железнодорожной автоматики и телемеханики.

## 6.1 Основные показатели безотказности и безопасности

*Безотказность СЖАТ* – это свойство системы непрерывно сохранять работоспособное состояние в течение некоторого времени или наработки [1]. Показатели безотказности делятся на две группы: показатели невосстанавливаемых и восстанавливаемых изделий.

*Невосстанавливаемые изделия* – изделия, поведение которых существенно лишь до первого отказа [11], – характеризуются следующими количественными показателями надежности: интенсивностью отказов $\lambda(t)$; вероятностью безотказной работы $P(t)$; вероятностью отказа $Q(t)$; средней наработкой до отказа $T_o$.

*Восстанавливаемые изделия* – изделия, эксплуатация которых допускает их многократный ремонт, – характеризуются следующими количественными показателями надежности: параметрами потока отказов $\omega(t)$ и потока восстановлений $\mu(t)$; функцией готовности $K_г(t)$; коэффициентом готовности $K_г$; средним временем работы между двумя отказами $t_{ср}$; средним временем восстановления $t_в$.

Если в процессе функционирования невосстанавливаемого изделия возможен ремонт отдельных его элементов при сохранении работоспособности изделия в целом за счет резерва или, если надежность функционирования восстанавливаемого изделия оценивается в интервале времени до первого отказа восстанавливаемого изделия в целом, то такие изделия характеризуются следующими количественными показателями надежности: вероятностью безотказной работы $P(t)$; вероятностью отказа $Q(t)$; наработкой до отка-

за $T_o$; параметрами потока отказов элементов изделия $\omega(t)$ и потока восстановлений элементов изделий $\mu(t)$.

*Безопасность СЖАТ* – это свойство системы непрерывно сохранять исправное, работоспособное или защитное состояние в течение некоторого времени или наработки [1]. Показатели безопасности аналогичны показателям безотказности.

Для невосстанавливаемых систем рассчитывают следующие показатели: вероятность безопасной работы $P_б(t)$; вероятность опасного отказа $Q_{оп}(t)$; интенсивность опасных отказов $\lambda_{оп}(t)$; среднюю наработку до опасного отказа $T_{оп}$.

Для восстанавливаемых систем рассчитывают следующие показатели: параметр потока опасных отказов $\omega_{оп}(t)$; функцию безопасности $K_б(t)$; коэффициент безопасности $K_б$; среднее время работы между двумя опасными отказами $t_{б.ср}$ [1].

Приведенные показатели безотказности и безопасности являются основными. Для реальных СЖАТ можно применять и другие дополнительные показатели, характеризующие структуру системы и ее конкретное назначение.

### 6.2 Методы расчета показателей безотказности и безопасности

Рассмотрим методы расчета показателей для различных структур, защищенных от опасных отказов. В этих структурах для сравнения результатов обработки информации на выходах и в контрольных точках вычислительных каналов используются компараторы с несимметричной характеристикой отказов.

Вероятность безотказной работы дублированной структуры с умеренной связью (рисунок 6.1) определяется выражением [12]

$$P_Д(t) = P_1^2(t) P_&^n(t) = e^{-\left(2\sum_{i=1}^{k}\lambda_i + n\lambda_&\right)t}, \qquad (6.1)$$

где $P_1$, $P_&$ – вероятности безотказной работы соответственно канала обработки информации и выходной схемы «И»;

$n$ – число информационных выходов модуля;

$\lambda_i$ – интенсивность отказов элементов канала обработки информации;

$\lambda_&$ – интенсивность отказов выходной схемы «И»;

$k$ – число элементов канала обработки информации.

Вероятность безотказной работы мажоритарной структуры с умеренной связью (рисунок 6.2) определяется выражением [12]

$$P_{2\vee 3}(t) = P_{\text{МЭ}}^{n}(t)\left[3P_1^2(t) - 2P_1^3(t)\right], \tag{6.2}$$

где $P_1$, $P_{\text{мэ}}$ – вероятности безотказной работы соответственно канала обработки информации и мажоритарного элемента МЭ;

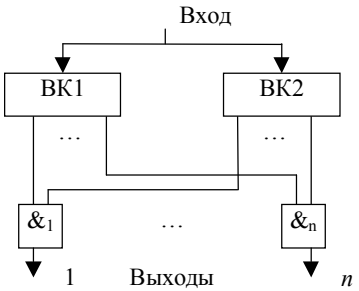



Рисунок 6.1 – Дублированная структура с умеренной связью

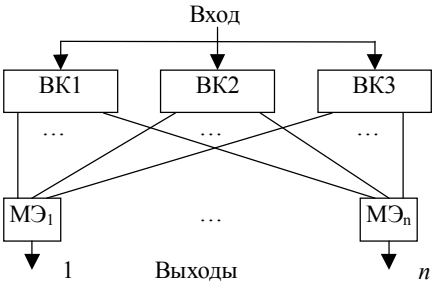



Рисунок 6.2 – Мажоритарная структура с умеренной связью

$n$ – число информационных выходов модуля.

Опасным отказом в резервированном микропроцессорном модуле считается отказ, необнаруживаемый встроенными средствами контроля. Учитывая очень малое значение интенсивности опасных отказов (для высоконадежных систем $\lambda_i\tau<<1$), вероятность появления опасного отказа в дублированных и троированных модулях определяется выражениями

$$Q_{\text{ОД}} \approx \frac{t}{\tau_{\text{Д}}}\left(\lambda_i\tau_{\text{Д}}\right)^2, \tag{6.3}$$

$$Q_{\text{О2}\vee 3} \approx 3\frac{t}{\tau_{\text{Д}}}\left(\lambda_i\tau_{\text{Д}}\right)^2, \tag{6.4}$$

где $\tau_{\text{Д}}$ – период диагностирования элементов модуля;

$\lambda_i$ – интенсивность отказов канала обработки информации.

При контроле идентичности работы каналов обработки информации по выходным сигналам трудно детерминировать поведение ЭВМ при появлении отказов, поэтому все возникающие отказы считаются эквивалентными. В этом случае оценка безопасности таких модулей получается несколько заниженной.

При введении в состав дублированного микропроцессорного модуля устройства контроля совпадения сигналов на шинах внутреннего интерфейса

(рисунок 6.3) показатели безотказности и безопасности определяются следующим образом [12]

$$P_{Д}^{КШ}(t) = P_1^2(t) P_{\&}^n(t) P_{УК}(t), \tag{6.5}$$

$$Q_{ОД}^{КШ} = \frac{t}{\tau_Д}\left[\sum_{i=1}^{k}(\lambda_i \tau_Д)^2\right], \tag{6.6}$$

где $P_1$, $P_{\&}$ – вероятности безотказной работы соответственно канала обработки информации и выходной схемы «И»;

$n$ – число информационных выходов модуля;

$P_{УК}$ – вероятность безотказной работы устройства контроля;

$\lambda_i$ – интенсивность отказов элементов канала обработки информации;

$k$ – число элементов канала обработки информации;

$\tau_Д$ – период диагностирования элементов.

Для дублированного модуля с устройствами контроля шин и выходов (рисунок 6.4) вероятность безотказной работы равна

$$P_{Д}^{КШ}(t) = P_1^2(t) P_{УК_1}(t) P_{УК_2}(t). \tag{6.7}$$

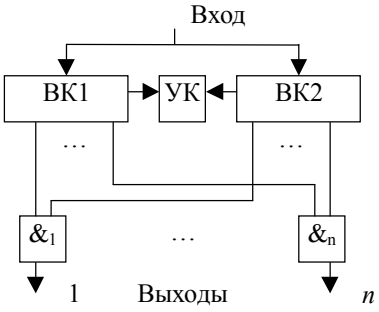



Рисунок 6.3 – Дублированная структура с устройством контроля шин

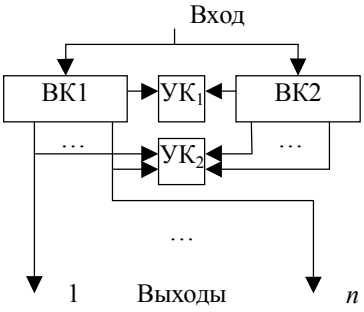



Рисунок 6.4 – Дублированная структура с устройствами контроля шин и выходов

Вероятность появления опасного отказа в этой структуре при условии, что $УК_1$ и $УК_2$ безопасны, определяется выражением (6.6).

Рассмотрим методы расчета показателей безотказности и безопасности в мажоритарных структурах. На рисунке 6.5 приведена структура мажоритарного модуля с попарным сравнением сигналов внутреннего интерфейса и выходными мажоритарными элементами, отвечающими требованиям безо-

пасности. Для этого случая показатели безотказности и безопасности определяются по формулам [12]:

$$P_{2\vee 3}^{\text{КШ1}}(t) = P_{\text{мэ}}^{n}(t)\left\{P_1^3(t)P_{\text{УК}}^3(t) + 3P_1^2(t)P_{\text{УК}}(t)\left[1 + P_1(t)P_{\text{УК}}(t) - P_{\text{УК}}^2(t) - P_1(t)\right]\right\}, \quad (6.8)$$

$$Q_{\text{ОД}}^{\text{КШ1}} = 3\frac{t}{\tau_{\text{Д}}}\left[\sum_{i=1}^{k}(\lambda_i\tau_{\text{Д}})^2\right], \quad (6.9)$$

где $P_1$, $P_{\text{мэ}}$ – вероятности безотказной работы соответственно канала обработки информации и выходной мажоритарной схемы;

$n$ – число информационных выходов модуля;

$P_{\text{УК}}$ – вероятность безотказной работы устройства контроля;

$\lambda_i$ – интенсивность отказов элементов канала обработки информации;

$k$ – число элементов канала обработки информации;

$\tau_{\text{Д}}$ – период диагностирования элементов.

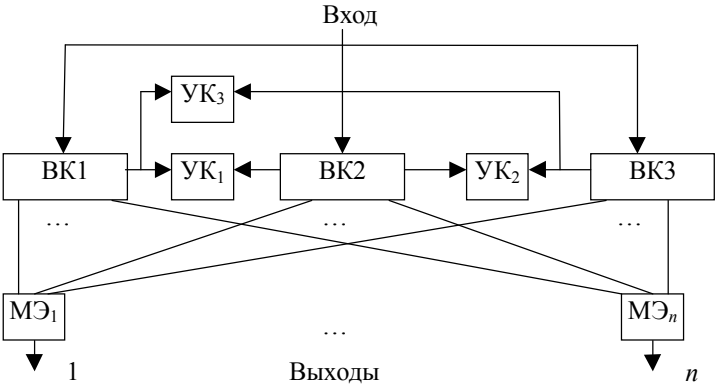



Рисунок 6.5 – Мажоритарная структура с попарным сравнением шин

На рисунке 6.6 приведена структура микропроцессорного модуля, в котором контроль идентичности работы каналов осуществляется путем сравнения сигналов на шинах ЭВМ с истинным значением, получаемым на выходе мажоритарного элемента. В этом случае вероятность появления опасного отказа определяется выражением (6.9), а вероятность безотказной работы такого модуля – выражением:

$$P_{2\vee 3}^{\text{КШ2}}(t) = P_{\text{мэ}}^{n}(t)P_{\text{мэк}}(t)\left[3P_1^2(t)P_{\text{УК}}^2(t) - 2P_1^3(t)P_{\text{УК}}^2(t)\right], \quad (6.10)$$

где $P_{\text{мэк}}(t)$ – вероятность безотказной работы мажоритарного элемента контроля (МЭК) за время $t$.

В этом случае вероятность безотказной работы сильно зависит от надежности МЭК. Уменьшить эту зависимость можно, используя три МЭК, как показано на рисунке 6.7. В этом случае

$$P_{2\vee 3}^{\text{КШЗ}}(t) = P_{\text{мэ}}^{n}(t)\left[3P_1^2(t)P_{\text{мэк}}^2(t)P_{\text{УК}}^2(t) - 2P_1^3(t)P_{\text{мэк}}^3(t)P_{\text{УК}}^3(t)\right]. \qquad (6.11)$$

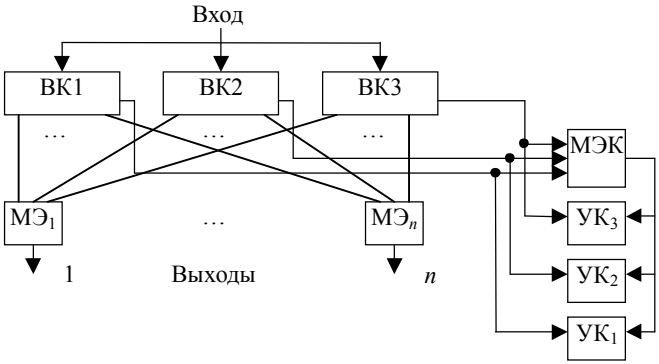



Рисунок 6.6 – Мажоритарная структура с мажоритарным элементом контроля

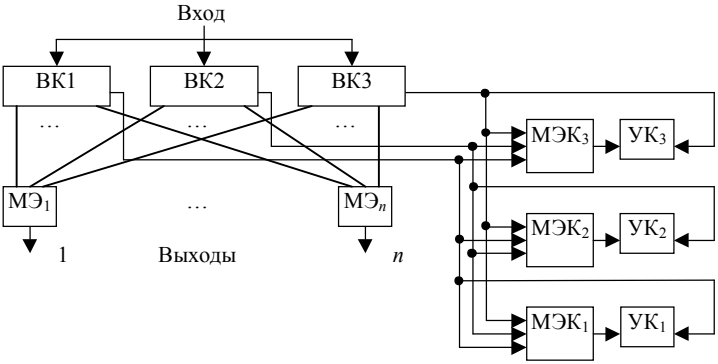



Рисунок 6.7 – Мажоритарная структура с резервированными мажоритарными элементами контроля

Таким образом, по выражениям (6.1) – (6.11) можно определить вероятности безотказной работы и вероятности появления опасного отказа для ос-

новных структур микропроцессорных централизаций. Полученные вероятности можно использовать для определения остальных показателей безотказности и безопасности, используя выражения взаимосвязи показателей.

### 6.3 Расчет показателей безопасности и безотказности микропроцессорных централизаций

Рассмотренные аналитические выражения для определения показателей безотказности и безопасности дублированных и мажоритарных структур могут использоваться и при расчетах более сложных структур. В этом случае для получения аналитического выражения применяется метод расчета, основанный на расчетно-логических схемах.

В качестве примера рассмотрим расчетно-логическую схему для определения показателей безотказности микропроцессорной централизации *SSI* (рисунок 6.8), где в качестве элементов схемы использованы: процессор интерфейса (ПИ), процессор централизации (ПЦ), процессор связи центральной подсистемы (ПС), модуль информационной связи (МИС), локальный компьютер (ЛК), компаратор, управляющий напольным устройством (Кр), схемы непосредственного управления объектами (СНУК).

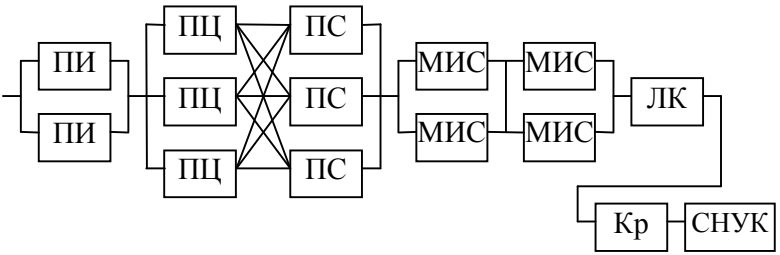



Рисунок 6.8 – Расчетно-логическая схема для определения показателей безотказности

В системе произведено мажоритарное резервирование ПЦ и ПС. Резервирование ПИ и МИС произведено по схеме постоянно включенного и нагруженного резерва. Из расчетно-логической схемы следует, что вероятность безотказной работы данной системы микропроцессорной централизации

$$P_{\text{о}}(t)=\left[1-\left(1-P_{\text{ПИ}}(t)\right)^2\right]\left[3P^2_{\text{ПЦ}}(t)-2P^3_{\text{ПЦ}}(t)\right]\left[3P^2_{\text{ПС}}(t)-2P^3_{\text{ПС}}(t)\right]\times$$
$$\times\left[1-\left(1-P_{\text{МИС}}(t)\right)^2\right]^2 P_{\text{ЛК}}(t)P_{\text{Кр}}(t)P_{\text{СНУК}}(t), \qquad (6.12)$$


7

где $P_{ПИ}$, $P_{ПЦ}$, $P_{ПС}$, $P_{МИС}$, $P_{ЛК}$, $P_{Кр}$, $P_{СНУК}$ – вероятности безотказной работы соответственно ПИ, ПЦ, ПС, МИС, ЛК, Кр, СНУК.

Для расчета показателей безопасности составляется расчетно-логическая схема, учитывающая только элементы, отказ которых может привести к опасному отказу системы в целом. Так как ПИ не влияет на безопасность системы, то его из расчетно-логической схемы можно исключить. Расчетно-логическая схема для определения показателей безопасности микропроцессорной централизации *SSI* приведена на рисунке 6.9.

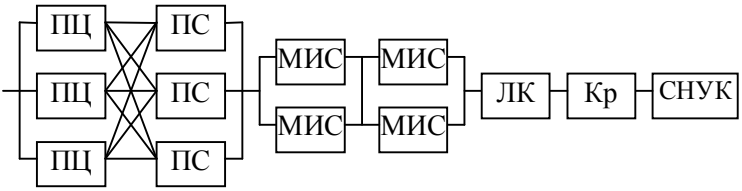



Рисунок 6.9 – Расчетно-логическая схема для определения показателей безопасности

Вероятность опасных отказов данной системы для случая $\lambda t << 1$ можно определить из следующего выражения:

$$Q_{оо} \approx 3Q_{ПЦ}^2 + 3Q_{ПС}^2 + 2Q_{МИС} + Q_{ЛК} + Q_{Кр} + Q_{СНУК}, \qquad (6.13)$$

где $Q_{ПЦ}$, $Q_{ПС}$, $Q_{МИС}$, $Q_{ЛК}$, $Q_{Кр}$, $Q_{СНУК}$ – вероятности опасного отказа соответственно ПИ, ПЦ, ПС, МИС, ЛК, Кр, СНУК.

На основании приведенных формул может быть произведен расчет времени наработки на отказ и наработки на опасный отказ системы микропроцессорной централизации или других показателей безопасности и безотказности.

# 7 ВНЕДРЕНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ ЖЕЛЕЗНОДОРОЖНОЙ АВТОМАТИКИ НА ЖЕЛЕЗНЫХ ДОРОГАХ МИРА

За последние 25 лет СЖАТ, построенные на базе микропроцессорной техники, постепенно вошли в железнодорожную практику. Их производством занимаются многие известные фирмы, а внедрение происходит не только в развитых, но и в развивающихся странах на участках, необорудованных СЖАТ, и на строящихся линиях.

Специалисты считают, что традиционные системы сигнализации, централизации и блокировки (СЦБ) обладают хорошей долговечностью (до 80 лет жизни) [1]. Поэтому темпы внедрения новых систем на железных дорогах обычно невысоки. К тому же эти темпы сдерживаются трудностями в решении проблемы безопасности микропроцессорных систем. В связи с этим на железных дорогах мира находится в эксплуатации много устройств и систем СЦБ разных поколений и модификаций. Например, в Швейцарии в настоящее время работают 163 механических, 260 электромеханических, 413 релейных и только одна микропроцессорная централизация стрелок и сигналов. Аналогично обстоит дело и во многих других странах.

В то же время существует устойчивая тенденция к разработке и внедрению микропроцессорных СЖАТ взамен выработавших свой ресурс старых систем. Переход на новую элементную базу обусловлен резко возросшими объемами работы и увеличением скорости обмена как управляющей, так и известительной информацией. Отдельные посты централизации охватывают участки до 250 км, на которых одновременно находятся сотни поездов, многие из которых движутся со скоростью свыше 250 км/ч [13].

Основные тенденции внедрения микропроцессорных СЖАТ рассмотрим на примере микропроцессорных систем централизации стрелок и сигналов.

Первая из систем нового поколения – микропроцессорная централизация (МЦ) *IZS-750* – была разработана шведской фирмой *ABB Signal* и внедрена на станции Гетеборг в 1978 г. Последующие системы этой фирмы (например, *Ebiloc 850*), а также системы других скандинавских фирм (например, система *IZSD 770* датской фирмы *DZI*) базируются на одноканальной структуре с

двумя диверситетными программами. Они работают в односекундном системном цикле с тестовым периодическим контролем. Для повышения надежности системы на больших станциях применяют второй компьютер в режиме горячего резерва. В настоящее время на железных дорогах Швеции, Дании, Норвегии, Финляндии, Испании, Болгарии и Польши работают более 100 централизаций данного типа.

Первая британская микропроцессорная централизация *SSI* (*Solid State Interlocking*) разработана фирмой *GEC/GB* и работает на станции Лимингстон Спа с 1985 г. Она построена по принципу однопрограммной многоканальной системы «два из трех» и функционирует в циклическом режиме с периодом 0,85 с. Связь с напольными объектами осуществляется модулями со структурой «два из двух». Аппаратные средства контроля подвергаются непрерывному тестированию. Структура системы допускает реконфигурацию при отказах. На разных станциях мира эксплуатируют более 20 подобных централизаций.

Фирмами *Westinghouse* (Великобритания), *Safetran* (США) и *Dimetronic* (Испания) разработана система микропроцессорной централизации *Westrace*, предназначенная для протяженных участков с раздельными пунктами малых размеров. *Westrace* – это система с одноканальным техническим и диверситетным программным обеспечением. Причем диверситетность программного обеспечения достигается различными методами компиляции одной и той же программы. Результаты работы программ сравниваются программным способом. Централизации этого типа внедрены на линиях местных перевозок в Австралии и Испании.

В разработке и производстве нового поколения систем в Германии принимают участие известные фирмы *Siemens, AEG* и *Alcatel SEL*. В 1983 году было принято решение об опытной эксплуатации систем микропроцессорной централизации, разработанных фирмами *Alcatel SEL, Siemens и AEG*.

Все три системы являются мультипроцессорными системами с тремя функциональными уровнями. Верхний уровень представлен ЭВМ, которые управляют вводом и отображением данных. ЭВМ среднего уровня (районные ЭВМ) реализуют сигнальные зависимости, обеспечивая формирование, замыкание и размыкание маршрутов, проверку допустимости воздействий на отдельные напольные устройства и т. д. Каждая ЭВМ этого уровня обслуживает определенный район станции. ЭВМ нижнего уровня непосредственно управляют напольными устройствами.

Базовая концепция фирмы *Siemens* получила название *SIMIS*. В ней используются два независимых вычислительных канала, которые работают синхронно и синфазно по одной программе с аппаратным сравнением по принципу «два из двух». Команды, вырабатываемые в каждом канале, срав-

ниваются двумя независимыми компараторами. Решения принимаются к исполнению только в тех случаях, когда сигналы на выходах обоих компараторов идентичны. В системе имеются тестовые программы, которые проверяют исправность вычислительных каналов в промежутках между обработкой оперативной информации. Безошибочность программного обеспечения достигается комплексными методами разработки и верификации.

Наряду с работой в режиме «два из двух» в особо ответственных ситуациях используются трехканальные безопасные структуры «два из трех», позволяющие повысить эксплуатационную готовность системы.

Вычислительные средства *SIMIS* имеют модульное построение. Это позволяет быстро формировать из них ту вычислительную структуру, которая соответствует конкретному объему решаемых задач и уровню требуемой надежности. Обмен информацией между компьютерами *SIMIS* осуществляется через двухканальную линию связи. В случае неисправности одного канала производится автоматическое переключение на одноканальный режим.

Существуют различные модификации систем, отличающиеся количеством управляемых объектов и требованиями безопасности. Система *EI S* предназначена для управления крупными станциями, *EI S Regio* – для малых станций, управляемых из единого центра, *SICAS* – для второстепенных участков и заводских путей.

В апреле 1989 года введена в эксплуатацию система микропроцессорной централизации на станции Кьяссо (Швейцария). В зону действия поста МЦ на станции Кьяссо входят 174 стрелки, 354 основных, маневровых и дополнительных сигнала, семь устройств путевой блокировки и 300 рельсовых цепей. До внедрения МЦ для управления перевозочным процессом на этой станции использовались один распорядительный и 4 исполнительных поста ЭЦ. МЦ базируется на микропроцессорных блоках компактного исполнения *SIMIS-C*. Всего в основном и двух вспомогательных зданиях поста МЦ установлено 69 ЭВМ.

Система микропроцессорной централизации *EI S*, установленная на станции Ганновер, осуществляет контроль и управление более чем 900 исполнительными устройствами.

Система *EI S Regio* предполагает автоматическое управление стрелками и сигналами участка железной дороги с общего поста диспетчерской централизации. В основу *EI S Regio* положен принцип минимального количества аппаратуры, обеспечивающей автоматическое управление движением поездов на станциях с минимальным путевым развитием. В состав минимального модуля напольной аппаратуры *EI S Regio* входят четыре главных сигнала, путевые датчики *Indusi*, два предупредительных сигнала, два стрелочных перевода с электроприводом, напольная аппаратура счета осей, устройство

запирания ключей от стрелочных замков. Этот минимальный модуль можно наращивать дополнительными элементами в зависимости от топологии станции.

Унифицированы также станционные модули, в которых устанавливается постовая аппаратура *EI S Regio*. Из станционного модуля можно управлять напольными устройствами, находящимися на удалении до 13 км.

Система *SICAS (Siemens Computer Aided Signalling)* предназначена для применения на региональных участках железных дорог, заводских путях, железнодорожных переездах, грузовых дворах. Это гибкая система централизации, удовлетворяющая менее высоким требованиям по безопасности и допускающая дальнейшее расширение путем добавления новых модулей. Предусмотрены исполнения системы *SICAS* с двумя или тремя вычислительными каналами. На уровне управления используются стандартные персональные ЭВМ, на базе которых строятся автоматизированные рабочие места (АРМ). Наличие стандартных компонентов и модульное построение позволяют минимизировать время проектирования системы и сократить затраты на разработку. Всего фирмой *Siemens* установлено более 100 различных типов микропроцессорных централизаций на железных дорогах Германии, Австрии и Швейцарии.

Микропроцессорная централизация типа *EIA*, разработанная фирмой *AEG* и находящаяся в эксплуатации на станции Дибург с 1988 г., также использует принцип многоканальной обработки информации. В МЦ фирмы *AEG* все ЭВМ, используемые для обработки ответственной информации, построены на базе универсальной микропроцессорной системы с безопасными отказами *LOGISIRE C*. Эта система состоит из двух идентичных вычислительных каналов, в которые загружено одинаковое программное обеспечение. Каналы работают независимо друг от друга. Безопасность функционирования обеспечивают специальная операционная система и защищенное от опасных отказов устройство контроля и отключения.

Микропроцессорная централизация *EI L* фирмы *Alcatel SEL* базируется на стандартных ЭВМ, на которых построены модули обеспечения безопасности *SELMIS*. Они имеют в своем составе контуры безопасности, являющиеся совокупностью аппаратных и программных средств. Принципы построения контуров безопасности позволяют использовать для выполнения ответственных команд стандартные ЭВМ, обладающие большой вычислительной мощностью.

Безопасность блоков SELMIS основана на многократной обработке информации в независимых параллельно работающих вычислительных каналах с последующим сравнением в этих же каналах входных, контрольных, промежуточных и выходных данных с помощью программных средств. Для

достижения высокой эксплуатационной готовности применяется преимущественно схема «два из трех».

Одним из перспективных направлений считается создание децентрализованных микропроцессорных систем управления. В Германии разработана децентрализованная система управления *MCDS*, построенная на базе устройства местного микропроцессорного управления стрелкой типа *IVV-MCEOW-2M*, дополненного устройствами управления светофором типа *IVV-MCSIG-2M* и контроля свободности пути типа *IVV-MCGFE2M*.

Децентрализованная ЭВМ может управлять группой из 32 устройств управления стрелками, 50 устройств контроля свободности пути и 50 сигналов. В систему *MCDS* можно объединить до 10 таких ЭВМ. Информацию, поступающую от децентрализованных ЭВМ и вводимую диспетчером, обрабатывает центральная ЭВМ. Текущая эксплуатационная ситуация отображается на экране цветного дисплея.

Центральная ЭВМ передает задания на установку маршрутов децентрализованным ЭВМ и выводит извещения о состоянии напольных устройств. Для ввода заданий служит клавиатура. Установку маршрутов осуществляют децентрализованные ЭВМ, в ПО которых реализованы все маршрутные зависимости.

Система микропроцессорной централизации *Elektra*, созданная фирмой *Alcatel Austria* совместно с Австрийским исследовательским центром в Зайберсдорфе, содержит два одинаковых в аппаратном отношении канала с различным программным обеспечением. Первый канал решает функциональные задачи и реализует логические зависимости при установке маршрутов. Второй канал осуществляет контроль безопасности. Программное обеспечение устройств логического канала составлено на языке программирования высокого уровня *CHILL*. Канал безопасности работает с экспертной системой *PAMELA*, в которой использован одноименный язык программирования, разработанный в исследовательском центре фирмы *Alcatel*. Экспертная система методами искусственного интеллекта на базе заложенных в нее знаний дежурного по станции и технологических инструкций осуществляет контроль правильности работы первого канала. Первая система МЦ Elektra была установлена в 1989 г. на станции Ноймарк-Кальхам Австрийских железных дорог.

Французская система МЦ разработана фирмой *Alsthom*. Система построена по модульному принципу, что позволяет учитывать специфику конкретной станции путем ввода соответствующих параметров в ПО каждого модуля. Для передачи информации между модулями служит кольцевая сеть. Разработка ПО МЦ ведется с помощью языка программирования высокого уровня *Ada*.

Американская фирма *General Railway Signal (GRS)* с начала 80-х годов разрабатывает и производит системы МЦ *VPI* с одноканальным техническим обеспечением. В стандартном исполнении система *VPI* может управлять 320 безопасными входами и выходами. Возможно объединение нескольких таких модулей для управления крупными станциями, такими как Нью-Йорк-Центральный, на которой работают совместно 17 систем *VPI*.

Безопасность при одноканальном техническом обеспечении в системе *VPI* достигается за счет того, что в центральной ЭВМ циклически одна за другой обрабатываются программа первичной логики и программа логики обеспечения безопасности. Программа первичной логики работает с дублированными данными, которые размещены в разных областях памяти. Обработка обоих массивов данных осуществляется последовательно одним и тем же программным обеспечением.

Главный цикл, реализующий логику централизации, имеет длительность 1 с. Он прерывается каждые 50 мс циклом обеспечения безопасности, в течение которого все включенные безопасные выходы проверяются на допустимость их включения в соответствии с массивами данных центральной ЭВМ. При обнаружении нарушения не позднее чем через 150 мс специальное реле размыкает цепь питания соответствующего модуля ввода-вывода.

Система достаточно широко эксплуатируется в США, Нидерландах, Испании, Италии, Австралии и странах Азии.

На железных дорогах Японии разработана и внедряется система МЦ *SMILE*. В нее заложена трехканальная структура с переменным порогом мажорирования, обеспечивающая высокий уровень эксплуатационной готовности. Процессоры работают синхронно от общего генератора тактов. Аппаратный компаратор с самоконтролем сравнивает потенциалы на внутренних магистралях попарно. Периодический контроль осуществляется с помощью программного счетчика. Для небольших станций применяется модификация *μSMILE*, двухканальная, с аппаратным самопроверяемым компаратором и горячим резервом. Первая система *SMILE* введена в постоянную эксплуатацию на станции Хигаси-Каганава в марте 1985г.

В литературе имеются сведения и о других современных микропроцессорных системах железнодорожной автоматики. В них используются принципы обеспечения безопасности, рассмотренные в данном пособии. Безотказность достигается использованием многоканальных систем. Обычно применяется трехканальное мажорирование или ненагруженный горячий (холодный) резерв. Безопасность достигается сравнением работы каналов с помощью самопроверяемых компараторов. Обеспечение безопасности однопрограммных систем возможно путем достижения безошибочности, а многопрограммных систем – усилением степени диверситета.

## СПИСОК ЛИТЕРАТУРЫ


1 Методы построения безопасных микроэлектронных систем железнодорожной автоматики / В.В. Сапожников, Вл. В. Сапожников, Х.А. Христов, Д.В. Гавзов; Под ред. Вл. В. Сапожникова. – М.: Транспорт, 1995. – 272 с.

2 РТМ 32 ЦШ 1115842.01-94. Безопасность железнодорожной автоматики и телемеханики. Методы и принципы обеспечения безопасности микроэлектронных СЖАТ. – СПб.: ПГУ ПС, 1994. – 120 с.

3 Балашов Е.П., Пузанков Д.В. Проектирование информационно-управляющих систем. – М.: Радио и связь, 1987. – 255 с.

4 Внедрение систем МЦ на железных дорогах Австрии // Железные дороги мира. – 1990. – №3. – С. 27–29.

5 Система микропроцессорной централизации фирмы AEG // Железные дороги мира. – 1990. – №5. – С.39–42.

6 *Linde H., Schiweck L. W. Der Sichrheitsnach weis auf Bauelementenebene// Signal und Draht,* 1981. – №9. – С. 17–21.

7 Майерс Г. Надежность программного обеспечения. – М.: Мир, 1980. –360 с.

8 Липаев В.В. Тестирование программ. – М.: Радио и связь, 1986. – 296 с.

9 Сертификация и доказательство безопасности систем железнодорожной автоматики / В.В. Сапожников, Вл. В. Сапожников, В. И. Талалаев и др.; Под ред. Вл. В. Сапожникова. – М.: Транспорт, 1997. – 288 с.

10 Харлап С. Н., Кочуров Д. С. Надежные программные реализации управляющих алгоритмов: Учебно-методическое пособие по подготовке к лабораторным работам по дисциплинам “Программно-математическое обеспечение микропроцессорных систем” и “Микропроцессорные информационно-управляющие системы на транспорте. – Гомель: БелГУТ, 1999. – 56 с.

11 Микропроцессоры: В 3 кн. Кн. 2. Средства сопряжения. Контролирующие и информационно-управляющие системы: Учебник для техн. вузов/ Под ред. Л.Н. Преснухина. – Мн.: Выш. шк., 1987. – 303 с.

12 РТМ 32 ЦШ 1115482.02-94. Безопасность железнодорожной автоматики и телемеханики. Методы расчета показателей безотказности и безопасности СЖАТ. – СПб.: ПГУ ПС, 1994. – 36 с.

13 Современные зарубежные системы микропроцессорной централизации (МПЦ) // Автоматика, связь, информатика – 2000. – №7. – С. 45–47.

14 Станционные системы автоматики и телемеханики: Учебник для вузов ж.-д. трансп. / Вл. В. Сапожников, Б.Н. Елкин, И.М. Кокурин и др.; Под ред. Вл. В. Сапожникова. – М.: Транспорт, 1997. – 432 с.

## Содержание

Учебное издание

*БОЧКОВ Константин Афанасьевич*,
*ХАРЛАП Сергей Николаевич*

**Методы обеспечения безопасности в микропроцессорных системах железнодорожной автоматики и телемеханики**

Учебное пособие

Редактор О. В. З а н и н а
Технический редактор В. Н. К у ч е р о в а
Корректор М. П. Д е ж к о

Подписано в печать 05.07.2001 г. Формат бумаги 60x84 $^{1}/_{16}$. Бумага газетная.
Гарнитура *Times New Roman.* Печать офсетная.
Усл. печ. л. 4,88. Уч.-изд. л. 4,94. Тираж 200 экз.
Зак. № 1446. Изд. № 3595.

Редакционно-издательский отдел БелГУТа, 246653, г. Гомель, ул. Кирова, 34.
Лицензия ЛВ № 57 от 22.10.97 г.

Типография БелГУТа, 246022, г. Гомель, ул. Кирова, 34.
Лицензия ЛП № 360 от 26.07.99 г.