С. С. РАСУЛОВА, А. А. РАШИДОВ

ПОСТРОЕНИЕ ОТКАЗОУСТОЙЧИВЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

С.С. РАСУЛОВА, А.А. РАШИДОВ

ПОСТРОЕНИЕ ОТКАЗОУСТОЙЧИВЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Рекомендовано Министерством высшего и среднего специального образования Республики Узбекистан в качестве учебного пособия для студентов высших учебных заведений

Ташкент—«Mehnat»—2004

32.973.26
P24

Изложены принципы обеспечения отказоустойчивости вычислительных систем путем организации эффективного контроля, тестирования и восстановления работоспособности компонентов. Рассмотрены модели надежности ОУВС на основе цепей Маркова, модели процессов контроля и восстановления ОУВС. Приведены примеры организации отказоустойчивых систем.

Учебное пособие предназначено для магистров, а также в помощь аспирантам и научным работникам, специализирующихся в одной из самых актуальных на сегодня областей – создании высоконадежных микропроцессорных систем.

Под редакцией доктора технических наук, профессора
Т.М. МАГРУПОВА

Рецензенты: доктор технических наук, профессор
Б.Б. АБДУСАТТАРОВ; доктор физико-
математических наук З.Т. АЗАМАТОВ

$P\frac{2302030700-36}{6\text{Б}39(04)-2003}$ без объявл. 2004

ISBN 5-8244-1593-5

ВВЕДЕНИЕ

В настоящее время проводится большая работа по увеличению объема производства средств вычислительной техники и повышению их надежности.

Для систем управления производственными процессами и оперативной обработки сообщений неуклонно повышаются требования к надежности результатов вычислений. Неисправность компьютерных систем телекоммуникаций, в управлении воздушным и наземным транспортом, на промышленных предприятиях, в банках, учреждениях, фондовых биржах ставит под угрозу всю их деятельность и влечет за собой значительные материальные потери, а нередко и человеческие жертвы.

В этих условиях особое значение приобретает развитие теории отказоустойчивых вычислительных систем (ОУВС).

Преимущества использования ОУВС очевидны, если учесть, что даже незначительные нарушения вычислительного процесса и сбои в управляющих вычислительных системах могут привести к крупным ошибкам в денежных расчетах, нарушению технологии производства продукции и т.п. Экономический эффект при использовании ОУВС обусловлен также снижением эксплуатационных расходов и увеличением времени непрерывной работы системы.

Сложность современных вычислительных средств такова, что практически невозможно проверить готовые изделия при всех предполагаемых условиях и режимах их работы. Поэтому в ОУВС могут быть скрытые — не проявившиеся при проверке ошибки программного обеспечения и (или) неисправности аппаратуры. Но благодаря достаточно высокому уровню отказоустойчивости сбои и отказы не

приводит к искажению выходных данных, отдельных элементов. ОУВС, основанные на различных принципах и вариантах технических решений, отличаются друг от друга степенью устойчивости к отказам и затрачиваемыми на это дополнительными (избыточными) аппаратными и программными средствами. Степень отказоустойчивости ВС зависит не только от количества избыточных средств, но и от способа организации их совместной работы в системе, режимов эксплуатации, стратегии ремонта и восстановления работоспособности.

Высокая надежность ОУВС позволит избежать потерь, обусловленных снижением производительности из-за возникновения ошибок, а также сократит затраты на обслуживание.

В самом общем плане для каждого конкретного случая использования ВС существует свое оптимальное решение, степень приближения к которому определяется полнотой и точностью информации о характеристиках системы и правильностью ее применения. Всякое другое решение приводит к потере производительности ВС и (или) к уменьшению степени достоверности получаемых результатов.

Существующие критерии и методы позволяют относительно быстро и точно оценить производительность ВС. Надежность электронных элементов ВС определяется в результате длительных и дорогостоящих экспериментов. Достаточно точные значения показателей надежности известны в основном только для выпускаемых промышленностью электронных элементов стабильной технологии и конструкции. Для новых разрабатываемых образцов эти показатели оцениваются приближенно.

Уровень надежности и производительности ВС определяются не только отказами отдельных элементов, сбоями, ошибками в программах, но и способами автоматического контроля, диагностирования, реконфигурации и восстановления данных, организации обслуживания ВС. Однако в настоящее время отсутствуют общие модели, учитывающие все эти факторы.

В настоящем пособии рассматриваются математические модели ОУИС, позволяющие в рамках имеющихся данных принимать решения по выбору и организации эксплуатации отказоустойчивых систем.

Авторы выражают большую благодарность В.В. Прощенко за помощь в разработке приведенных алгоритмов и моделей отказоустойчивости.

Глава 1. ОСОБЕННОСТИ ПРОЦЕССА ПРОЕКТИРОВАНИЯ ОТКАЗОУСТОЙЧИВЫХ СИСТЕМ

1.1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Последние достижения в области повышения надежности средств вычислительной техники характеризуются появлением вычислительных систем (ВС), способных нормально функционировать даже при отказах отдельных ее компонентов. Такое свойство ВС получило название «отказоустойчивость», а сами ВС — «отказоустойчивые вычислительные системы» (ОУВС).

Разработка средств обеспечения отказоустойчивости ВС неразрывно связана с применением терминов и понятий, используемых при рассмотрении общих и частных вопросов надежности технических объектов. Поэтому приведем ряд установленных ГОСТ 27.002—93 терминов и основных понятий, которые будут часто встречаться в тексте.

Вычислительная система (ВС) — совокупность программных и аппаратных средств преобразования информации, объединенных некоторой формой регулярного взаимодействия с целью выполнения рабочего задания.

Архитектура ВС — совокупность свойств системы, воспринимаемых пользователем.

Вычислительный процесс — совокупность координированных действий системы, необходимых для выполнения рабочего задания.

Надежность ОУВС — свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих ее способность выполнять требуемые функции в заданных режимах и в условиях возникновения отказов и сбоев отдельных ее компонентов.

Долговечность ОУВС — свойство системы сохранять работоспособность до наступления предельного состояния при установленном режиме технического обслуживания и ремонта.

Предельное состояние ВС — такое состояние системы,

после наступления которого ее дальнейшая эксплуатация технически невозможна или экономически нецелесообразна.

Отказ компонента ВС — событие, заключающееся в нарушении работоспособности компонента ВС.

Сбой — самоустраняющийся отказ компонента ВС, приводящий к кратковременной утрате его работоспособности.

Ошибка — проявление сбоя или отказа компонента ВС.

В работах, посвященных надежности и отказоустойчивости вычислительных систем, часто понятия толерантности, живучести и отказоустойчивости отождествляются. Будем употреблять понятие «отказоустойчивость».

Отказоустойчивость — свойство системы, позволяющее продолжить выполнение заданных программой действий после возникновения одного или нескольких сбоев или отказов компонентов ВС.

d-устойчивость ВС — устойчивость системы относительно числа отказов компонентов ВС, не превышающего d.

Отказ ОУВС — событие, заключающееся в нарушении функционирования системы из-за сбоя или отказа компонента, при котором выполнение рабочего задания не может быть продолжено с помощью имеющихся в ВС программных и аппаратных средств автоматического восстановления в течение максимально допустимого времени простоя.

Конфигурация ВС — совокупность и способ взаимодействия программных и аппаратных средств системы, направленных на выполнение целевой функции—рабочего задания.

Реконфигурация ОУВС — изменение состава и способа взаимодействия программных и (или) аппаратных средств системы с целью исключения отказавших программных или аппаратных компонентов.

Восстановление ОУВС — автоматическое восстановление работоспособности системы.

Ремонт ОУВС — восстановление работоспособности системы с помощью специалиста.

Избыточность ВС — дополнительные программные и (или) аппаратные средства, возможности алгоритма или

время для выполнения дополнительных вычислений, предназначенные для повышения надежности ОУВС.

Если алгоритм способен обеспечить правильный результат, несмотря на возможные отдельные ошибки в ходе вычислений, то говорят об *алгоритмической избыточности*. К таким алгоритмам относятся, например, алгоритмы, основанные на методе статистических испытаний (методе Монте-Карло). Если обрабатываемое сообщение содержит некоторое повторение информации в той или иной форме, которое позволяет восстанавливать исходную информацию в случае каких-либо нарушений в работе системы, то принято говорить об *информационной избыточности*.

Характерным способом введения избыточности является резервирование — использование дополнительных средств и (или) возможностей с целью сохранения работоспособности ВС при отказе одного или нескольких ее элементов.

Различают *статическую* и *динамическую* избыточности. Статическая избыточность реализуется автоматически сразу после того, как произойдет отказ. Система построена таким образом, что после отказа ее ненарушенная часть позволяет продолжить выполнение задания. При этом нарушение «маскируется» (не проявляется). Динамическая избыточность реализуется только после некоторой перестройки работы системы, получившей сигнал об отказе устройств контроля.

Основными показателями контролепригодности (ГОСТ 27.002—83) объектов являются следующие:

— коэффициент полноты проверки исправного (работоспособного) состояния $K_{пп} = \lambda_к/\lambda_0$, где $\lambda_к$ — суммарная интенсивность отказов проверяемых составных частей объекта на принятом уровне деления; λ_0 — суммарная интенсивность отказов всех составных частей системы на принятом уровне деления;

— коэффициент глубины поиска дефекта $K_{гп} = F/R$, где F — число однозначно различимых составных частей системы на принятом уровне деления, с точностью до которых определяется место дефекта; R — общее число составных частей системы на принятом уровне деления, с точностью до которых требуется определение места дефекта;

— длина теста диагностирования $L = \{1,2,3,... |L|\}$, где $|L|$ — число тестовых воздействий.

Показатели диагностирования:

Вероятность ошибки диагностирования вида $P_{i,j}$, $i \neq j$.

$P_{i,j}$ — вероятность совместного наступления двух событий:

система по диагностированию находится в техническом состоянии i, а в результате диагностирования считается находящейся в состоянии j.

Вероятность правильного диагностирования — полная вероятность, что система диагностирования определяет то техническое состояние, в котором действительно находится объект диагностирования.

Комплексные показатели надежности:

Коэффициент готовности — вероятность того, что система окажется в работоспособном состоянии в произвольный момент времени, кроме планируемых моментов, в течение которых применение системы по назначению не предусматривается.

Коэффициент оперативной готовности — вероятность того, что система окажется в работоспособном состоянии в произвольный момент времени, кроме планируемых моментов, в течение которых применение системы по назначению не предусматривается, и, начиная с этого момента, будет работать безотказно в заданном интервале времени.

Коэффициент сохранения эффективности — отношение значения показателя эффективности за определенную продолжительность эксплуатации к номинальному значению этого показателя, вычисленному при условии, что отказы системы в течение такого же периода эксплуатации не возникают.

Основным средством обеспечения отказоустойчивости ВС является резервирование.

1.2. ОРГАНИЗАЦИЯ ОТКАЗОУСТОЙЧИВОСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Характерной особенностью ОУВС является то, что свойство отказоустойчивости реализуется за счет автоматического устранения влияния ошибок в вычислениях, происходящих из-за отказов элементов системы, на конечный

результат той или иной задачи. Это достигается введением в систему дополнительных средств (избыточности) аппаратного и программного обеспечения, которые назовем средствами обеспечения отказоустойчивости (СОО). При этом следует иметь в виду, что сущность процесса проектирования СОО остается такой же, как и при проектировании средств, предназначенных непосредственно для обработки информации пользователя. Тем не менее сложность процесса проектирования таких систем значительно возрастает за счет того, что они должны автоматически выполнять ряд специфических функций, в общем случае скрытых от пользователя. Кроме того, на реализацию указанных функций влияют как характеристики решаемых задач, так и случайные величины, связанные со временем и местом возникновения ошибок в вычислениях.

Такими функциями являются следующие [1, 3, 5]:

1. Обнаружение факта возникновения ошибки в вычислениях.

2. Исправление ошибки в вычислениях.

3. Распознавание сбоев и отказов аппаратуры системы.

4. Локализация отказов элементов системы.

5. Определение текущего состояния системы по состояниям ее элементов.

6. Принятие решения по изменению алгоритма функционирования системы при наличии отказавших элементов.

7. Перестройка структуры системы (в простейшем случае автоматическая замена отказавшего элемента резервным).

8. Восстановление отказавшей аппаратуры (возможно с участием человека).

Все эти функции, с точки зрения их функционального содержания можно разделить на две группы. Первая состоит только из одной функции – обнаружение факта возникновения ошибки в вычислениях, вторая содержит все остальные, общее назначение которых – восстановление работоспособного состояния системы. Поэтому в дальнейшем мы будем первую группу функций называть *функцией обнаружения*, а вторую – *функцией восстановления*.

Как показывает опыт построения ОУВС, затраты на стадии их разработки и изготовления дополнительных ап-

[illegible] и программных средств СОО могут быть значи[illegible]. Однако в процессе их применения они окупаются за счет снижения затрат на приобретение запасных ком[illegible], которые необходимы в системах, не обладающих свойством отказоустойчивости, а также за счет [illegible] общих эксплуатационных издержек за срок службы системы, благодаря почти полному устранению простоев из-за отказов элементов и различных профилактических мероприятий.

Необходимо также иметь в виду, что в современных и перспективных областях применения ВС имеются случаи, [illegible] принцип отказоустойчивости должен быть реализован обязательно. К ним относятся следующие случаи применения:

— [illegible] отказ будет угрожать жизни человека (управление [illegible] поездами, самолетами, пилотируемыми космическими кораблями и другими транспортными средствами; в управлении атомными электростанциями, системами оборонных и контроля и жизнеобеспечения в автоматизированных службах больниц и т.д.

— [illegible] даже короткий простой ВС может повлечь за собой тяжелые экономические последствия (управление производством и технологическими процессами на автоматизированных предприятиях, распределением электроэнергии, системами коммутации и телефонии и других видах [illegible], в условиях, исключающих возможность ручного обслуживания (беспилотные космические корабли, станции контроля за состоянием природной среды в удаленных и труднодоступных районах и т.д.).

Процесс проектирования СОО условно можно представить в виде последовательности этапов, на которых решаются следующие задачи.

1. **Определяется целевая функция обеспечения отказоустойчивости системы.** На этом этапе анализируется техническое задание на разработку системы, в том числе определяется целевое назначение системы (например, управление технологическим процессом), выясняются условия эксплуатации (особенности окружающей среды, возможности восстановления аппаратуры, режимы функционирования), анализируются особенности и характеристики основных алгоритмов. Затем четко определяются классы

неисправностей, по отношению к которым необходимо обеспечить устойчивость ко всем вообще видам отказов в системе, и, наконец, определяются требуемые количественные критерии отказоустойчивости. Данные критерии должны давать возможность достаточно полно оценить при способленность системы к выполнению заданных функций и следовательно, влиять на объективно правильный выбор варианта ее организации при проектировании.

2. Разработка архитектуры системы с учетом реализации функций СОО. На данном этапе задачей является определение закона функционирования системы при сбоях и отказах на основе установленной ее целевой функции. В соответствии с этим законом последовательно решаются вопросы выбора и введения в систему алгоритмов обнаружения ошибок, локализации отказов, восстановления информации и аппаратуры.

Выбранный метод реализации функции обнаружения ошибок определяется установленными показателями отказоустойчивости и должен обеспечивать своевременное выявление всех требуемых классов неисправностей. Разработка алгоритмов восстановления, которые запускаются сигналами от алгоритмов обнаружения ошибок, ведется с учетом основной функции — возврату системы к нормальному режиму обработки информации или безопасному останову.

Выбор методов восстановления определяется ожидаемым уровнем повреждения аппаратуры и информации и требуемой скоростью восстановления нормальной работы. Следует также учитывать возможность вмешательства в процесс восстановления извне по отношению к отказавшему компоненту системы (цифровой системы или человека). Введение рассматриваемых функций может потребовать изменений в первоначальном варианте архитектуры системы за счет добавления схем и программ контроля, резервных процессорных или шинных элементов, увеличения емкости памяти, изменения последовательностей управляющих сигналов и т.д.

Если функции обнаружения, локализации отказов и восстановления информации могут быть реализованы за счет специальной организации автономных компонентов системы, то функции определения текущего состояния

принятия решения по изменению алгоритма функционирования и управлению перестройкой системы могут быть реализованы только как общесистемные функции, следовательно, они обязательно должны учитываться при идеологической проработке архитектуры системы. Следует отметить, что существуют способы введения избыточности, которые минимизируют реализацию функций обнаружения ошибок, локализации отказов и восстановления информации, например, мажоритарное резервирование со схемами голосования, рассматриваемое ниже.

3. **Оценка отказоустойчивости системы.** Данная задача может быть решена с помощью аналитического моделирования или комбинацией обоих методов. На этом этапе важно не только получение численных значений показателей характеристичности системы, но и выявление так называемых «узких» ее мест, т.е. таких компонентов, которые вносят наибольшую долю в снижение значений рассматриваемых показателей.

4. **Уточнение архитектуры разрабатываемой системы на основе полученных результатов моделирования.**

Все перечисленные задачи при проектировании ВС следует рассматривать не как требование автономности их решения, а как элемент процесса создания вычислительной системы, который обусловлен единым функциональным ее назначением — получать и выдавать достоверные результаты обработки данных за заданное время с заданными показателями качества. И с этих позиций процесс проектирования ВС рассматривается как процесс реализации взаимосвязанных функций обработки информации в СОО с заданными показателями качества и при заданных ограничениях.

1.3. ХАРАКТЕРИСТИКИ НАДЕЖНОСТИ КОМПОНЕНТОВ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Прежде чем приступить к проектированию отказоустойчивой системы, разработчик должен оценить надежность ее элементов. Допустим, в качестве элемента системы выбран некоторый мини-ЭВМ. Простейшая оценка надежности объекта исследования заключается в определении интенсивности его отказов λ_0 как суммы интенсивностей отка-

зов компонентов объекта и затем вычислений средней наработки его до отказа $T_{ср}$. При экспоненциальном распределении наработка до отказа $T_{ср}=1/\lambda_0$.

В табл. 1 [5] приведен пример расчета интенсивности отказов для процессорной части типичной мини-ЭВМ.

Таблица

Расчет интенсивностей отказов для процессорной части типичной мини-ЭВМ

Компонент	Интенсивность отказов компонентов 10^{-6}	Центральный процессор		ЗУ на магнитных сердечниках, емкость 4 К слов		Панель управления		Источники питания и распределительные шины	
		Число компонентов	Интенсивность отказов	Число компонентов	Интенсивность отказов	Число компонентов	Интенсивность отказов	Число компонентов	
1	2	3	4	5	6	7	8	9	10
ИМС	0,1	190	19,0	70	7,00	20	2,00	10	1,0
Диод	0,02	6	0,12	150	3,00	–	–	10	
Диод	0,5	–	–	–	–	–	–	10	
Транзистор	0,05	–	–	60	3,00	–	–	–	
Транзистор	0,3	–	–	–	–	–	–	9	2,7
Конденсатор	0,002	120	0,24	150	0,30	–	–	15	0,0
Конденсатор	0,04	–	–	–	–	–	–	3	0,1
Резистор	0,01	160	1,60	300	3,00	80	0,80	100	1,0
Резистор	0,1	–	–	–	–	–	–	2	0,2
Трансформатор	0,1	1	0,10	40	4,0	–	–	–	
Трансформатор	0,2	–	–	–	–	–	–	3	0,6

Окончание табл. 1

1	2	3	4	5	6	7	8	9	10
	0,0001	4000	0,40	3000	0,30	400	0,04	200	0,02
	1,4	4	14,00	–	–	–	–	–	–
	1,0	–	–	4,00	–	–	–	–	–
	0,00001	–	–	64 К	0,64	–	–	–	–
	0,2	–	–	–	–	25	5,00	–	–
	0,5	–	–	–	–	50	25,00	–	–
	0,1	–	–	–	–	–	–	2	0,20
	0,5	–	–	–	–	–	–	1	0,50
	0,3	–	–	–	–	–	–	1	3,00

В этих расчетах фигурируют числа компонентов в каждом устройстве и интенсивность отказов компонентов, вычисленные с учетом таких факторов, как окружающая температура, степень нагрузки и т.д. Следует отметить, что суммирование интенсивностей отказов компонентов дает оценку надежности снизу, так как все отказы компонентов приведут к останову процессора или выдаче неверного результата. Например, большинство возможных неисправностей пульта управления связано с отказами индикационных ламп, которые не скажутся на правильности вычислительного процесса.

Интенсивность отказов всей процессорной части для двух модификаций и средняя наработка ее до отказа показаны в табл. 2.

Интенсивность отказов остальной части оборудования мини-ЭВМ двух различных модификаций и ее суммарный показатель надежности приведены в табл. 3. В этой таблице НМД – накопитель на магнитных дисках, НМЛ – накопитель на магнитной ленте.

Таким образом, подобный расчет надежности элементов ЦВМ позволяет получить исходные данные для создания средств обеспечения их отказоустойчивости.

Таблица

Интенсивность отказов процессорной части и средняя наработка её до отказа

Устройства	Процессор с памятью 4 К слов		Процессор с памятью 16 К слов	
	Число устройств	Интенсивность отказов $\times 10^{-6}$	Число устройств	Интенсивность отказов $\times 10^{-6}$
Центральный процессор	1	35,46	1	35,46
ЗУ на магнитных сердечниках	1	25,24	4	100,96
Пульт управления	1	32,84	1	32,84
Источник питания	1	14,37	1	14,37
Вся процессорная часть		107,91		183,63
Средняя наработка до отказа, ч		9267		5446

Таблица

Интенсивность отказов остальной части оборудования мини-ЭВМ и ее суммарный показатель надежности

1	2
Мини-ЭВМ А	Интенсивность отказов $\times 10^{-6}$
Процессор с памятью 4 К слов	107,91
Контроллер пишущей машины	10,20
Пишущая машинка	1000,00
Всего	1118,11
Средняя наработка до отказа	894 ч
Мини-ЭВМ	Интенсивность отказов $\times 10^{-6}$
Процессор с памятью К слов	163,63
Дополнительная память 48 К слов	302,88
Контроллер пишущей машины	10,20
Пишущая машинка	1000,00
Контроллер НМД	15,40

Окончание табл. 3

1	2
[illegible]	256,00
[illegible]	14,30
[illegible]	345,00
[illegible]	10,50
[illegible]	250,00
[illegible]	13,20
[illegible]	420,00
[illegible]	14,37
[illegible]	2843,48
[illegible]	352 ч

1.4. ОСНОВНЫЕ ЭТАПЫ ПРОЕКТИРОВАНИЯ ОУВС

Реализация программ создания отказоустойчивых систем занимает в среднем 5–15 лет и зависит от уровня промышленного освоения новых элементов и устройств микроэлектронной и вычислительной техники.

При [illegible] планировании разработок сложных систем можно выделить следующие стадии [3, 14]:

1. Анализ, теоретическая и экспериментальная проверка систем;

2. Разработка перспективных систем с учетом упреждения морального и технического старения;

3. Модификация существующих и разработка новых вариантов систем с целью удовлетворения новых требований к технико-экономическим параметрам системы.

Первая стадия часто сопровождается разработкой экспериментальных моделей и систем, что составляет в среднем 3–7 лет, включая техническое и рабочее проектирование, разработку документации и изготовление опытного образца.

Вторая стадия завершается отработанной до готовности к сдаче в опытную эксплуатацию базовой системой, которая используется для последующих ее модификаций. Сроки создания систем колеблются в широких пределах (3–10) лет в зависимости от качества начальных разработок, пра-

вильности выбранных принципов построения, стабильности и производительности основного контингента работчиков, возможностей выхода на промышленное дрение системы и т.д.

Третья стадия разработки системы обычно проводится параллельно с эксплуатацией базовой системы и позволяет улучшить ее технико-экономические показатели.

Вторая стадия во многих случаях обеспечивает основную эффективность и отдачу от применения системы, поскольку в конечном счете при правильной организации работы обеспечивает получение значительно лучших параметров и характеристик, охраноспособность и новизну системы с учетом сроков морального старения.

На стадии модификации разрабатываются новые варианты системы, совершенствуется обслуживание и корректируются технические решения по результатам эксплуатации и в соответствии с новыми требованиями. Изменение технологии изготовления, создание более совершенного программного обеспечения, определение режимов наиболее эффективного пользования системы могут служить важными факторами существенного повышения ее производительности и надежности.

На второй стадии разработки отказоустойчивой микропроцессорной системы можно выделить следующие этапы:

1. Определение целей введения отказоустойчивости и значений технических показателей системы, например, повышение коэффициентов технического использования и готовности системы на заданном интервале времени, сокращение затрат на обслуживание, увеличение вероятности безотказного выполнения системой ответственных задач.

2. Формирование требований к структуре связей и ее организации в системе с учетом особенностей использования технических средств.

3. Определение состава аппаратных и программных средств системы, корректировка решений, полученных на втором этапе.

4. Техническое проектирование с разработкой конкретных вариантов системы и моделирование ее работы.

5. Работы проектирование и создание опытного образ-
ца с последующей передачей в промышленную эксплуата-
цию.

Выполнение работ на каждом из перечисленных этапов
тесно связано со структурой и особенностями орга-
низации физических и логических связей в системе. На-
пример, выбор элементов системы с заданным интерфей-
сом взаимодействия элементов накладывает ограничения
на скорость обмена данными. Последнее служит основой
для расчетов времени функционирования элемента по
выполнению диагностических и рабочих функций.

На первом этапе определяются основные требования к
отказоустойчивым системам, среди
которых можно выделить следующие:

1. Защищенность системы от влияния аварийных неис-
правностей элементов.

2. Симметричность организации средств встроенного
контроля (СВК).

3. Возможность поиска неисправных элементов при ма-
лом количестве связей.

4. Наличие средств обеспечения перестройки структуры
системы для исключения различных неисправностей.

5. Унифицированность программ диагностирования не-
исправностей. Требования к реализации СВК при аварий-
ных ситуациях («зависание» системы, непредсказуемое до-
стояние основных линий связи и др.) состоит в обес-
печении процесса поиска неисправных элементов в системе
при отказе элементов, участвующих в диагностировании.
Обычно для защиты системы от аварийных неисправнос-
тей используются различные способы информационного и
структурного резервирования [15].

Симметричность СВК вытекает из удобства использо-
вания таких систем. В частности, симметричность СВК по-
зволяет унифицировать алгоритмы диагностирования эле-
ментов и системы в целом.

Наличие большого количества связей в СВК может
привести к увеличению времени диагностирования, повы-
шению аппаратурной сложности реализации, усложнению
синхронизации и обмена информацией между отдельны-
ми элементами. Поэтому поиск дефектов в СВК с наимень-

шим количеством связей приводит к снижению ди
мого количества диагностируемых неисправных эле
или увеличению длительности проверки и восстано
системы. Разумный компромисс между количеством
зей и заменяемых элементов системы обеспечивает
тивное диагностирование.

Перестройка структуры системы является одним из
ных резервов обеспечения живучести системы и восст
ления ее работоспособности.

Используя перестройку структуры, можно обеспе
одновременное диагностирование одних элементов и
боту по основному назначению других элементов сис

Требование к унификации и простоте алгоритмов
агностирования обусловлено тем, что соответствующие
программы должны распределяться по элементам сис
и работать в неблагоприятных условиях, при наличии
шого количества элементов.

Отказоустойчивые системы даже при отказе боль
количества элементов позволяют избежать мгновенн
выхода их из строя. Наблюдается лишь постепенное у
шение (деградация) их функциональных характери
что отличает ОУВС от обычных систем, в которых по
работоспособности происходит даже при отказе одно
элементов. Параллелизм диагностирования и функци
рования достигается при некотором снижении произ
тельности системы.

Построение эффективности перестраиваемых СВК
жно основываться на анализе возможных вариантов и в
чает решение следующих задач [30]:

1. Выбор структуры связей элементов системы, обе
чивающей диагностирование и выполнение системой
новных функций. Вид структуры зависит от требовани
качеству проверки и производительности системы при
шении основных задач.

2. Определение перечня элементов, входящих в с
СВК, из числа свободных от функционирования. В про
се работы выбираются те из них, которые обеспечи
выполнение поставленных задач контроля и диагности
вания.

3. Оценка характеристик эффективности функци

Накопление таких характеристик дает ... использовать потенциальные возможности и ... системы и своевременно перестроить процесс ... и диагностирования системы таким образом, ... необходимую ее отказоустойчивость.

... ВЫБОР СТРУКТУР ВЗАИМОКОНТРОЛЯ

При решении задачи построения структур взаимоконт... необходимо учитывать различные параметры проце... диагностирования неисправностей в системе. В зависи... от избранного способа диагностирования средства ... контроля (СВК) различаются количеством и ... связей элементов, временем и стоимостью вос... системы и т.п. В конечном счете это сказыва... на общей стоимости обслуживания системы и ее про... Так, например, при диагностировании с ... требуется большое количество связей вза... между элементами, однако необходимо боль... количество шагов диагностирования, замены и вос... работоспособных элементов в системе по ... диагностированием без восстановления. При ... без восстановления используется лишь ... поиска неисправностей и восстановления ... системы, но время выполнения проце... может быть достаточно большим [3, 12].

Поскольку увеличение времени диагностирования и ... приводит к возрастанию стоимости обслу... снижению производительности системы, то при ... СВК необходимо проводить оценку схемы по па...

Пусть каждая из структур взаимоконтроля характеризу... набором параметров $\{b_{ij}\}$. Параметры b_{ij}, $i = 1...n$, ... может, например, кратность обнаруживаемых не... при диагностировании с восстановлением и ... число резервных элементов, обеспечивающ... отказоустойчивость системы и т.д.

Каждому варианту l диагностирования и восстановле... сопоставить вектор-столбец $\{C_{ij}\}$, $l = 1...m$, эле... которого соответствует «стоимость» или «вес» ... параметра j из подмножества $\{b_{ij}\}$. Так,

например, кратности обнаружения неисправности в
тегии 1 может соответствовать определенное время
неисправных элементов системы.

Рассмотрим матрицу D, каждый элемент которой
ределяется как

$$d_{k,t} = \sum_{j=1}^{m} b_{k,j} \cdot C_{j,i},$$

т.е. D есть произведение матриц

$$B = \{b_{i,j}\} \text{ и } C = \{c_{j,t}\}, \text{ т.е. } D = BC.$$

Каждая строка матрицы D соответствует «взвешенно
оценкам соответствующей СВК при различных страте
диагностирования и восстановления.

Введем некоторую общую оценку $d_{j,0}$ строки j матри
D. В качестве такой оценки могут выступать, напри
максимальные, минимальные приведенные относите
некоторого значения элементов строки. Значение $d_{j,0}$
ветствует количественной оценке качества структуры j
различных стратегий диагностирования и восстановл
из множества заданных.

Обозначим через $d_{0,i}$ некоторую количественную ме
для столбца j матрицы D. Она будет характеризовать ка
ство стратегии i для множества заданных структур.

В зависимости от задаваемых требований, используя
и $d_{0,j}$, можно рассчитать показатели функционирова
системы для подмножеств структур и стратегий. Вводя пр
ничные значения $d^1_{0,i}$, $d^1_{0,j}$ можно отбросить заведомо бе
перспективные варианты структур и стратегий. Это позв
ляет выбирать наилучшие варианты одновременно на мно
жестве структур и стратегий, в наибольшей степени
удовлетворяющих всем требованиям.

Рассмотрим пример оценки СВК с использованием
изложенного способа. Предположим, что каждая структура
S_j характеризуется кратностью обнаруживаемых неиспра
ностей $b_{j,1}$ числом $b_{j,2}$ элементов, при которых система сч
тается работоспособной, количеством допустимых сос

Пусть имеются четыре стратегии i ($i = 1, 2, 3, 4$) … вариации из которых характеризует значение … параметра C_j = 1...3 структур S_j. Так, … для стратегии $b_{i,j}$ соответствует определенное … обработки работы, значению $b_{i,2}$ соответствует … снижения производительности системы (выражается в единицах времени), а $b_{i,3}$ — время обработки … времени с учетом вероятности восстановления, … матрицы D характеризуют качество откажущей системы, выраженное в единицах времени. … вероятности

$$B = \begin{vmatrix} 2 & 2 & 30 \\ 3 & 1 & 20 \\ 3 & 2 & 10 \end{vmatrix},$$

$$C = \begin{vmatrix} 1 & 1 & 2 & 3 \\ 0,2 & 0,3 & 0,1 & 0,2 \\ 0,01 & 0,02 & 0,02 & 0,01 \end{vmatrix},$$

тогда $D = BC = \begin{vmatrix} 2,7 & 3,2 & 4,8 & 4,7 \\ 3,4 & 3,7 & 6,5 & 6,4 \\ 3,5 & 3,8 & 6,4 & 6,5 \end{vmatrix}$.

… выборе в качестве оценки минимальных простоев

$$d_{1,0} = \min_{j=1,\,2,\,3,\,4} d_{1,j} = 2,7;$$

$$d_{2,0} = \min_{j=1,\,2,\,3,4} d_{2,j} = 3,4;$$

$$d_{3,0} = \min_{j=1,2,\,3,4} d_{3,j} = 3,5$$

… что наилучшей является первая структура, … $d_{1,0} = 2,7$.

… аналогично $d_{n,j}$, получаем, что наилучшей

является первая стратегия диагностирования и восстановления. Если ввести новое значение

$$d^1_{i,j} = \begin{cases} 0 \text{ при } d_{i,j} \geq 3{,}5 \\ 1 \text{ при } d_{i,j} < 3{,}5 \end{cases},$$

то получаем модифицированную матрицу

$$D' = \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}.$$

Нулям соответствуют неудовлетворительные стратегии и структуры отказоустойчивой системы. Поэтому «усеченная» матрица

$$D'' = \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix}$$

описывает подмножество структур и стратегий обслуживания отказоустойчивой системы, удовлетворяющих заданным временным характеристикам. В частности, удовлетворительными считаются две структуры и стратегии диагностирования и восстановления. Необходимо отметить, что с помощью матриц D′, D″ можно решать задачи расчета и оптимизации вариантов технических решений отказоустойчивых систем.

Для проведения более полного анализа необходимо строить недетерминированные модели процесса контроля и диагностирования, которые позволяют проводить более полную оценку на основе использования методов моделирования и статистических характеристик работы системы.

С учетом изложенного выше можно предложить следующий обобщенный алгоритм построения структур отказоустойчивых систем [2, 9]:

1. Выбрать и оценить различные виды структур взаимоконтроля по количеству связей, числу обнаруживаемых неисправностей.

2. Выбрать и оценить временные и аппаратурные затра-

допустимых для данных видов
алгоритмов поиска неисправностей.

Оценить время восстановления элементов системы для
алгоритмов поиска неисправностей.

Выбрать способ построения, алгоритмы поиска и вос-
приводящие к минимальным затратам по
устойчивости системы. Если получение
невозможно, то при заданных ограниче-
на некоторые значения одних характеристик следует
минимальных затрат по другим характеристикам.

Если не все варианты структур рассмотрены, то пе-
в пункт 1, изменив количество неисправных элемен-
аварийных неисправностей системы.

Провести корректировку структуры взаимоконтроля
полученной структуры для снижения
на восстановление системы.

Исходя из требований диагностируемости, удается по-
большое разнообразие структур взаимоконтроля,
по сложности реализации связи, возмож-
использования алгоритмов диагностирования. По-
в процессе диагностирования результаты конт-
могут существенно упростить структуру взаимоконт-
если удается локализовать значительную часть
элементов, либо, наоборот, могут привести
необходимости расширения количества связей с целью
неисправностей (в частности, аварийных).

Таким образом, возникает задача быстрой перестройки
структуры систем. Зная, в частности, наибольшую крат-
отказов, удается построить СВК, обеспечивающие
неисправностей для группы или всех элемен-
системы.

При оценке систем удобно использовать аналитичес-
способы построения СВК, обеспечивающие получе-
вариантов СВК с малыми затратами памяти и времени
помощью простых алгоритмов. Рассмотрим один из таких

Система S описывается D_{δ_1} – структурой, если связь
от элемента y_i к элементу y_j существует только в
случае, когда

$$j - i = \delta\, m(\mathrm{mod}\ n),\ m = 1,\ldots, t. \qquad (1.5.1)$$

Система, для которой структура $D_{\delta, t}$ удовлетворяет условию взаимной простоты чисел и δ и t, является $n = |V|$ — диагностируемой без восстановления t-ДС. Следовательно, при заданном значении количества элементов n=|V| существует множество СВК, удовлетворяющих требованию t-ДС.

Исходя из соотношения (1.5.1), получение допустимых структур взаимоконтроля можно проводить перечислением всех простых чисел δ и n. Поскольку все графы G = G(V, E) при фиксированных взаимно простых числах δ и n являются изоморфными между собой, т.е. существует взаимооднозначное соответствие между вершинами графов, в котором сохраняются соотношения инцидентности вершин множества V, то из множества допустимых структур выбирают СВК с наименьшими показателями стоимости и наибольшими показателями эффективности. В соответствии с выражением (1.5.1) алгоритм получения t-диагностируемых без восстановления СВК имеет следующий вид:

1. Принять δ равным 0.

2. Вычислить δ = δ + 1.

3. Если δ = n, то это конец операции. Все структуры перечислены. Если δ < n, то следует перейти к следующему шагу.

4. Определить наибольший общий делитель чисел δ и n.

5. Если наибольший делитель (общий) больше единицы, то надо перейти к шагу 2. В противном случае перейти к следующему шагу.

6. Вычислить и выдать структуру взаимоконтроля, удовлетворяющую заданным значениям δ, t и n.

7. Перейти к шагу 2. Шаг 4 может быть выполнен, например, с использованием алгоритма Евклида, а шаг 6 алгоритма — перебором допустимых связей при заданных δ, t и n.

На рис. 1 *а* показан общий вид предложенного алгоритма, а на рис. 1 *б* — реализация, соответственно, алгоритма Евклида для шага 4 и для шага 6 построения структур. Здесь через d обозначен наибольший делитель для δ и n, а $c = r(A/B)$ — остаток от деления A на B.

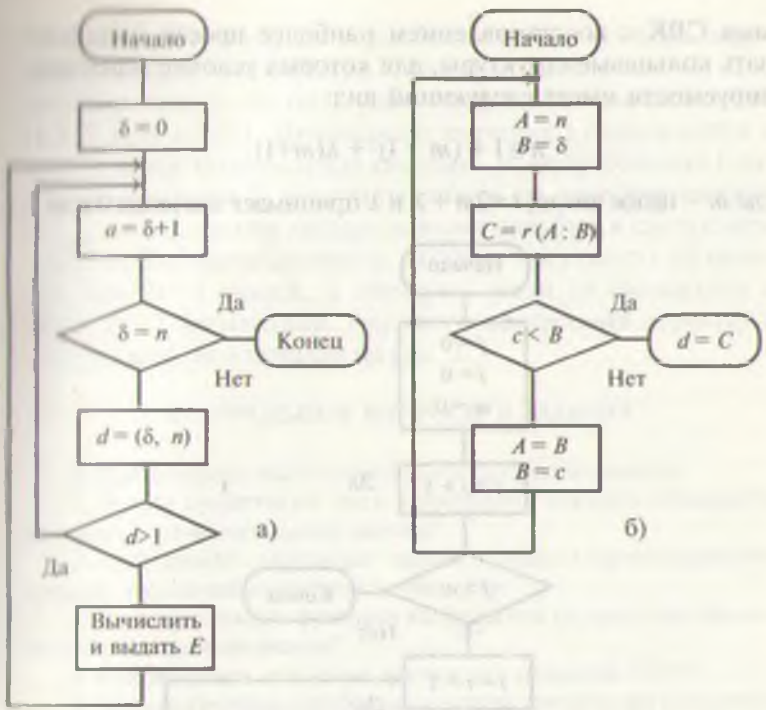


Рис. 1. Алгоритмы построения структур без восстановления (а) и Евклида (б)

Простота алгоритма обеспечивает быстрое построение СВК, при проектировании системы и при перестройке по результатам диагностирования и восстановления в процессе эксплуатации. В ряде случаев оказывается целесообразно строить СВК с ограниченным числом связей.

Обозначим наибольшее количество связей, которые могут иметь элементы v_1 v_2,..., $v_n \subset V$, входящие в V, а выходящие — через d_j^+, тогда СВК могут быть построены первоначально для

$$d_{max}^+ = \max d_i^+ \; u \; d_{min}^- = \min d_j^- \qquad (1.5.2)$$
$$j = 1,..., n, \qquad j = 1,..., n$$

с последующим дополнением оставшихся связей между остальными элементами. При построении t-диагностируе-

мых СВК с восстановлением наиболее просто использовать кольцевые структуры, для которых условие t-диагностируемости имеет следующий вид:

$$n \geq 1 + (m + 1)^2 + \lambda(m+1),$$

где m – целое число; $t = 2m + \lambda$ и λ принимает значение 0 или 1

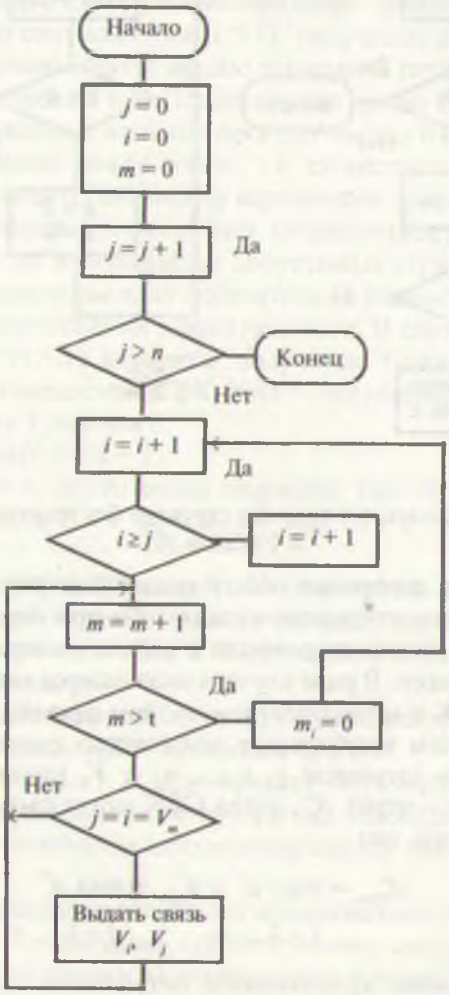


Рис. 2. Алгоритм построения структур с восстановлением

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

10. Оцените средства встроенного контроля ОУВС с использованием метода, описанного в п. 1.5.

Глава 2. МЕТОДЫ ИССЛЕДОВАНИЯ И ПОСТРОЕНИЯ ОУВС

2.1. ОБЗОР СУЩЕСТВУЮЩИХ ОУВС

В настоящее время одним из перспективных прикладных и научно-исследовательских направлений развития вычислительной техники является создание отказоустойчивых систем, предназначенных для обработки сообщений в реальном масштабе времени.

Отказоустойчивые вычислительные системы используются для управления технологическими и производственными процессами, системой банковских операций, системами резервирования мест в средствах транспортировки и передвижения, гостиничных номеров и др.

Применение ОУВС позволяет значительно повысить производительность процессов управления различными сферами деятельности предприятий, так как при этом обеспечивается постоянное поступление необходимой текущей информации.

Эффективность использования ВС определяется количеством и характером решаемых ею задач, поэтому, с одной стороны, чем шире круг задач, решаемых системой, тем выше эффективность ее использования, а с другой стороны, отказ такой ВС ведет к значительным материальным потерям. В этом смысле представляет интерес ОУВС, которые могут непрерывно (или в течение гораздо большего времени, чем обычные ВС) вести расчеты или управлять объектами. Они позволяют изолировать отказы отдельных компонентов системы и сохранять необходимую базу данных. Однако эта изоляция не может быть абсолютной. Отказоустойчивые системы с высокой степенью вероятности гарантируют, что отказ будет обнаружен и локализован, а затем будут восстановлены база данных и вычислительный процесс.

Отказоустойчивость системы оценивается вероятностью, с которой она гарантирует автоматическое восстановление

отказов отдельных компонентов, ...ности от отказов, а также «покры... количеством отказов компонентов, ...ется имение специалистов.

...к достижению высокой отказоус... в 70-х годы было использование избыточности ... резервирования отдельных подсистем, ... значительного увеличения количе... Кроме того, органы управления системы ... элементы, отказ которых мог привести к ... системы.

...ных фирм последнее десятилетие веду... разработки и создания ОУВС является Tandem ... Inc., которая финансирует разработки таких ком... как Stratus Computer Inc., Synapse Computer, ... В 1975 г. фирма Tandem предложила новый ... к обеспечению отказоустойчивости, ре... в ВС Tandem NonStop. При этом была реше... ... восстановления системы после отказа при ее ра... в реальном масштабе времени. Отказавшие компонен... могут быть удалены и возвращены после ремонта в ... без предварительных изменений в программном ... Все это стало возможным благодаря трем осо... ... организации работы системы [8,17]:

1. Дублированию процессоров;
2. Дублированию входов устройств управления вводом-...
3. Сохранению основной (базовой) операционной сис... многократно копируемой в памяти отдельных про...

Система Tandem NonStop включает от 2 до 16 микро... МП, которые могут функционально заменять ... друга. Каждый из них имеет собственную память объе... 8 Мбайт и канал ввода-вывода. Эти процессоры со... ... дублированной системой высокоскоростных па... ... шин Dynabus. Каждое устройство управления ... выходом имеет два канала связи и доступно для ... процессоров. Накопители на магнитных дисках (НМД) ... имеют два канала связи, каждый из которых соеди... ... устройствами управления.

Таким образом, база данных доступна даже в том [illegible] чае, когда и процессор и устройство управления НМД [illegible] казали. При отказе база данных может быть восстановлена, если на другом диске сохранились все необходимые данные. После отказа система автоматически восстанавливает две одинаковые копии рабочих файлов на двух независимых НМД. Затем восстанавливается остальная информация, необходимая для продолжения работы системы.

На рис. 3 изображена структура системы Tandem NonStop, состоящая из трех процессоров *1*, четырех устройств управления вводом-выводом *2*, имеющих по два канала связи *5* и *6*, и параллельной дублированной системы шин *3*, *4*.

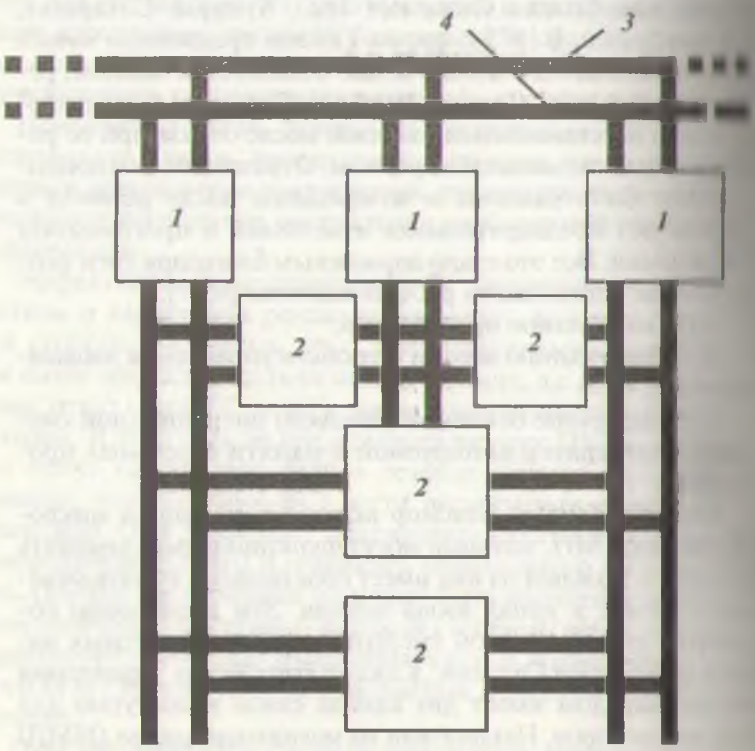


Рис.3. Структурная схема системы Tandem NonStop

[illegible] этой ВС имеет свою собственную [illegible] операционной системы Guardian, а также таблицу, [illegible] ресурсы системы на текущий момент. Процессоры периодически ежесекундно оповещают друг друга [illegible] сообщении, передавая сообщения «я жив» по шине [illegible]. Если такие сообщения не поступают от какого-[illegible] процессора, то остальные изменяют свои таблицы [illegible], исключая процессоры, не пославшие сообще-[illegible].

Основным механизмом восстановления в системе [illegible] NonStop является использование контрольных точек. [illegible] рабочий (основной) процесс имеет идентичный по [illegible] процесс (процесс «поддержки») в другом процессоре. При нормальном функционировании основной процессор в очередной контрольной точке передает своему «дублеру» информацию о своем состоянии и состоянии вычислительного процесса, после чего продолжает выполнение задания до следующей контрольной точки. В случае отказа рабочего процессора его функции берет на себя процессор «поддержки». Он вместе с операционной системой подводит «итог» работы основного процесса (куда выполнения задания) и продолжает вычисления, начиная с последней контрольной точки, в которой информация не была разрешена.

Несмотря на внешнюю простоту, реализация такого [illegible] — создание специального программного обеспечения — весьма сложна, поэтому в настоящее время он используется только для внутренних компонентов системы [illegible] NonStop, т.е. для операционной системы. Для внешних компонентов (прикладных программ) применяется другая схема, по которой восстановление происходит путем возвращения к более ранней контрольной точке или даже путем повторного выполнения задания.

Процесс пользователя (прикладная программа) благодаря системным сообщениям изолирован от изменяющейся конфигурации системы. Если этому процессу требуются некоторые данные с диска, то он формирует сообщение, которое обрабатывается локальной копией операционной системы (ОС) Guardian, а она, в свою очередь, по таблице ресурсов определяет путь доступа к НМД. Процесс

пользователя не решает, от какого из двух процессоров, соединенных с НМД, он получает ответ или какой из них будет играть роль «лидера». Это позволяет производить постепенное наращивание системы путем добавления процессоров.

Система Tandem NonStop занимает лидирующее положение среди ОУВС реального времени. Одним из интересных принципов построения ОУВС является контроль дублированием, воплощенный в системе Stratus (Stratus Computer Inc.). В этой системе каждая основная функция выполняется четыре раза объединенными по входам четырьмя независимыми процессорами. Устройства сравнения генерируют сигнал ошибки в случае расхождения двух результатов в одной из двух пар. Пара процессоров с несовпадающими результатами отключается, а другая пара продолжает работу. При подключении отремонтированной пары процессоров необходима синхронизация с дублирующей парой. Такой подход — «спаривание пар» — имеет два преимущества:

1. Не требуется времени для автоматического восстановления после отказа, т.е. работа продолжается без задержки. Короткое прерывание необходимо только при подключении отремонтированного процессора для синхронизации его работы.

2. Поскольку не требуется восстановление информации, следовательно, нет необходимости использовать контрольные точки.

Отметим, что в этой системе, как и в любой ОУВС, существуют внутренние средства диагностики и восстановления.

Система *Stratus 32* характеризуется довольно большой избыточностью. Ее основным недостатком является значительное усложнение базового устройства, которое в этой системе называется процессорным модулем. Система может насчитывать до 32 микропроцессоров (МП) (объемом памяти 8 Мбайт каждый), соединенных попарно кольцом локальной сети. Один процессорный модуль состоит из 4 МП типа Motorola 68000 и содержит свою копию ОС VOS.

Три четверти ресурсов системы являются резервными и поэтому не повышают ее производительность. [17, 24]

продолжает фирмой Synapse Computer
предложила микропроцессорную систему
, построенную на МП типа Motorola 68000.
процессоры взаимодействуют через дублированную
параллельную шину и имеют общую разделен-

На рис. 4 изображена структурная схема этой системы,
из общей памяти 1, в которой хранятся про-
системных заданий и заданий пользовате-
, а также списки заданий (очередность работ) из уни-
процессора 2 и параллельной системы шин 3, 4.
может содержать до 28 МП. Одни процессоры пред-
для операции ввода-вывода, другие — для об-
процессорных сообщений. Сами процессоры выс-
в качестве диспетчеров, обращаясь к общей памяти
выбирая задания. Операционная система
в общесистемной памяти объемом 16 Мбайт.

В название системы Sunapse входит выражение N + 1,
что дополнение к N работающим процессорам еще од-
обеспечивает такую же отказоустойчивость системы,
в системах ОУВС с 211 процессорами, т.е. в схемах,
каждый процессор «дублируется на дублера». При этом
работоспособных процессоров продолжает выпол-
задание неисправного процессора, используя инфор-

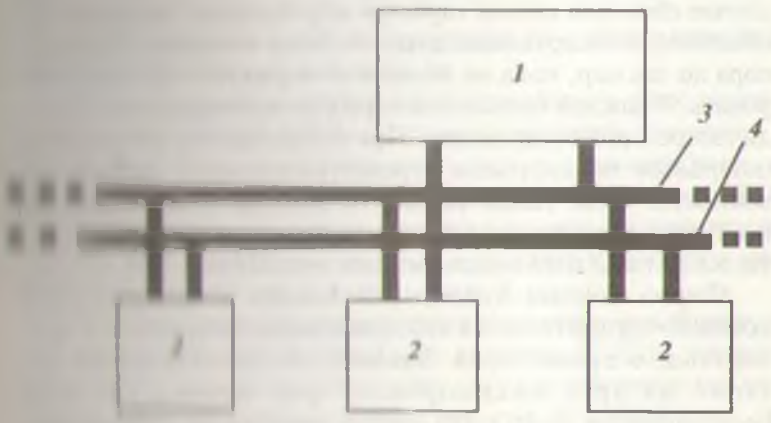


Рис. 4. Структурная схема системы Sunapse N+1

мацию, хранящуюся в общей памяти системы. Данный подход позволяет повышать производительность системы в такой же степени, в какой растет число ее процессоров (в отличие, например, от системы типа Stratus/32).

Однако общая разделенная память является «узким местом» в смысле отказоустойчивости. Существуют два способа, позволяющие преодолеть этот недостаток. Первый основан на отделении отказавшего модуля памяти и реализован в системе Synapse. Второй заключается в дублировании памяти системы и реализован в процессоре 3B20D Bell Laboratories Western Electric.

Интересное решение предложила фирма Auragen Systems Corp. Она разработала Systems 4000, в которой процессорные элементы соединены высокоскоростной дублированной шиной в так называемую «гроздь». Каждая «гроздь» объемом памяти 8 Мбайт состоит из трех МП типа Motorola 68000. Два из них выполняют задание пользователя, а третий реализует функции ОС. Каждая «гроздь» содержит свою копию операционной системы Unix Systems III.

В Systems 4000 использован принцип контрольных точек (точек синхронизации). После выполнения части задания происходит синхронизация: дублирующий процессор получает и сохраняет все сообщения, посланные основным процессором, а также запоминает пути следования этих сообщений с момента последней синхронизации. В случае сбоя или отказа «дублер» обрабатывает входные сообщения, блокируя выходные сигналы основного процессора до тех пор, пока он не начнет нормально функционировать. В каждой подсистеме «грозди» производится периодическое самотестирование. При его успешном завершении остальным подсистемам передается сигнал о работоспособности. Если такой сигнал не вырабатывается или не поступает к другим подсистемам, то данная «гроздь» считается остальными подсистемами отказавшей [14].

Фирма Tolerant Systems Inc. создала ОУВС Plus 32, в которой осуществляется синхронизация основного и дублирующего процессоров. Базовый элемент структуры состоит из двух микропроцессоров фирмы National Semuconductor Corp's NS 16000, связанных дублированной локальной сетью Elite Ethernet. Каждый из микропро-

операционной системы и
память до 1 Мбайт. Диагностирование
также, как в системе Sunapse N+1,
времени выполнения отдельных опера-
сигнала работоспособности.
Telecom разработана ОУВС Power S/55,
восьми микропроцессоров Motorola 68000
до 4 Мбайт.
обеспечивает полную взаимосвязь между любой
процессоров, имеющих свою копию ОС Perpos, и
всеми устройствами управления НМД. Это значи-
повышает вероятность изоляции одного из компо-
МП, НМД или части системы. Так как информа-
на содержание базы данных, хранится
НМД, значительно сокращается время поиска и
информации. Откат процессора или НМД диаг-
по превышению времени выполнения опера-
В этом случае любой из работающих МП и НМД благо-
связанности может продолжить выполне-
Наиболее общими чертами всех описанных ОУВС явля-
1. Максимально попарная связанность отдельных ком-
высокоскоростными дублированными
2. Наличие копии ОС в памяти каждого процессора или
процессоров подсистемы.
3. Возможность контроля работоспособности и восстанов-
системы при работе в реальном масштабе времени.
4. Использование универсальных, а следовательно, и
микропроцессоров типа Motorola 68000.
5. Разбиение основного процесса на ряд подпроцессов
количеству МП подсистем), в каждом из которых ис-
принцип дублирования.
Получили также распространение дублированные и
троированные вычислительные системы [3,7,30], напри-
полностью дублированная синхронизированная си-
AS 220H и троированная система AS 220HF (рис. 5),
контрольными органами (≤2) снабжены отдельно

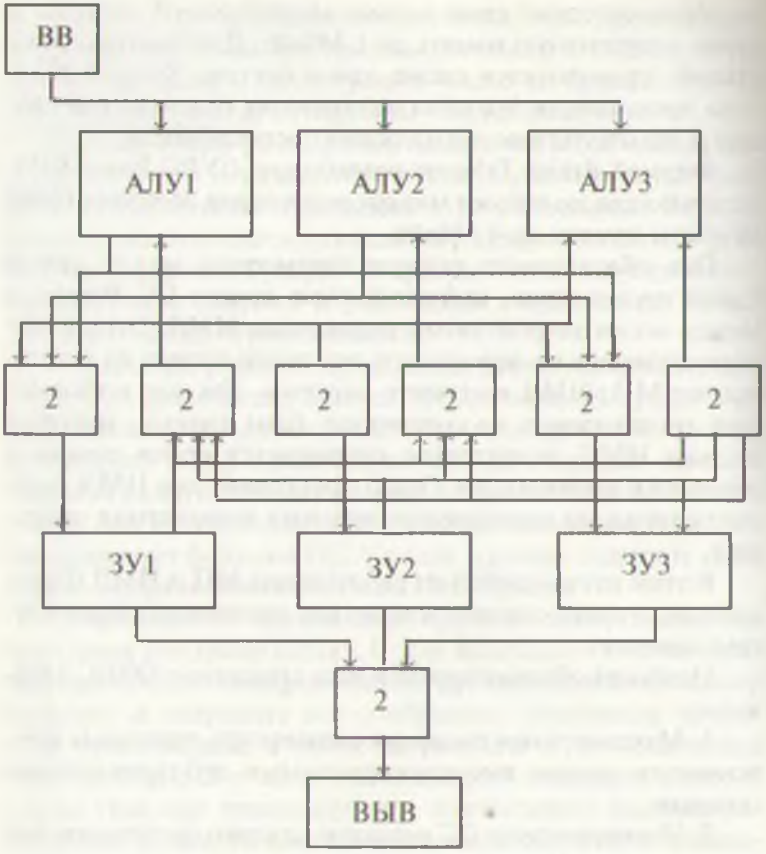


Рис.5. Структурная схема системы AS 220HF

арифметико-логические устройства (АЛУ), запоминающие устройства (ЗУ) и устройства ввода (ВВ) и вывода (ВЫВ). Базой этих систем является нерезервированная управляющая ВС AS 220.

Поскольку асинхронные ВС не позволяют быстро обнаружить отказы и ошибки, а также восстановить вычислительный процесс, в системе AS 220H используется синхронный принцип работы, позволяющий обнаружить отказ за несколько миллисекунд. В системе реализован принцип разделения функций обнаружения и локализации отказов.

Для полного представления об уровне развития ОУВС приведем основные данные современных зарубежных ВС (см. табл.4).

В таблице значения вероятности безотказной работы даны с учетом резервирования. В случае дублирования ВС эти данные соответствуют интенсивности отказов порядка 10^{-3} $ч^{-1}$ для одной ВС при использовании компонентов наиболее высокого качества и надежности. Как правило, дублированы устройства ввода-вывода, центральный процессор и устройства управления памятью. Оперативные запоминающие устройства объемом в 16 К слов секционированы (количество секций размером в 6 К слов выбирается по необходимости). Устройства для включения резерва основаны на логических схемах с переплетениями. Здесь очень большое значение имеет контроль и диагностирование, поскольку по их результатам вырабатываются сигналы для автоматического включения резерва.

Таблица 4

Основные технические данные и характеристики современных зарубежных ВС

Характеристики	Тип ВС					
	CDC 469	DELCO M362	GEDEC POP-11	LIT-TON 4516E	RCA SCR-234	Pockwell DF-224
1	2	3	4	5	6	7
Быстродействие, тыс.оп/с	200	650/840	100	300	70	400
Длина слова, Бит	16	16/32	16	16/32	16	24
Фиксированная Ф или плавающая П запятая	Ф	Ф/П	Ф	Ф/П	Ф	Ф
Объем оперативного ЗУ, К слов	64	64	32	128	64	64
Длительность цикла, мкс	1,6	0,6	—	—	2,34	1,6
Характеристика ввода—вывода, мкс/слов	2,5	2,0	1,0	2,0	16,6	4,8

Окончание табл. 2

1	2	3	4	5	6	7
Потребляемая мощность, Вт	20	150	14	80	5	100
Масса, кг	4	17,7	10	10,8	8	44
Размеры, дм, дм^3	1,61× 1,36× 1,55	20,8	3,05× 2,03× 50,8	8,2	5,9	4,57× 4,73× 3,23
Интервал допустимых рабочих температур, °С	−35... +60	−54... +86	−35... +70	−35... +70	−5... +45	−20... +50
Вероятность безотказной работы	0,925 за 1 год	0,99 за 2 года	0,99 за 2 года 0,95 за 3 года	0,98 за 2 года	0,899 за 2 года	0,92 за 3 года

2.2. АНАЛИЗ ПРИНЦИПОВ ПОСТРОЕНИЯ ОУВС

2.2.1. Основные задачи создания ОУВС

Области применения СВТ с каждым годом интенсивно расширяются. В связи с этим актуальной становится проблема создания новых вычислительных систем. Перед разработчиками ВС стоят две основные задачи:

– достижение высокой производительности;

– обеспечение высокой надежности.

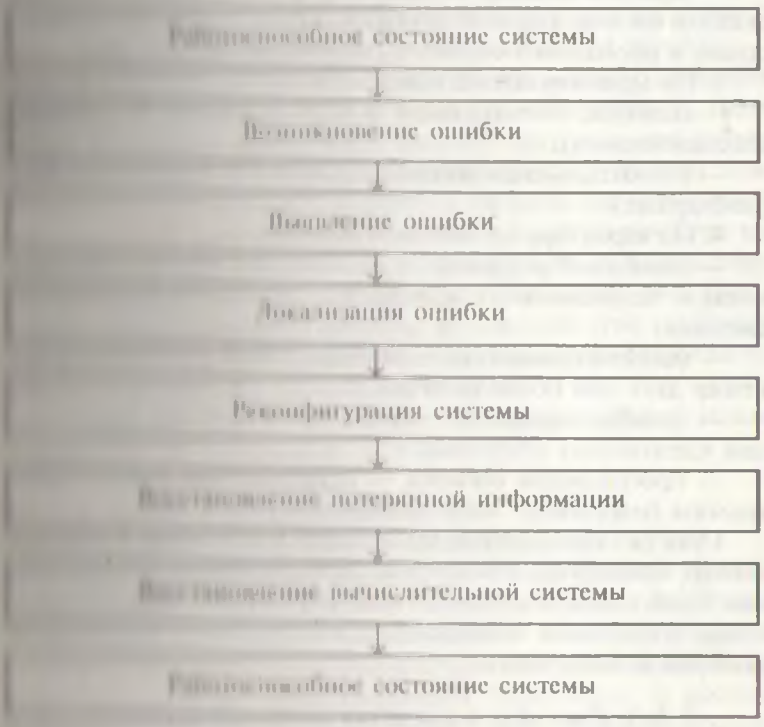
Эти задачи противоречивы, и в каждом конкретном случае необходимо выбирать компромиссное решение. Путь решения первой задачи — повышение быстродействия отдельных элементов ВС и максимальное распараллеливание вычислительного процесса. При решении второй задачи возможны два основных подхода [11,24]:

1. Предотвращение отказов системы реализуется путем повышения технологического уровня изготовления компонентов ВС (т.е. повышение надежности отдельных частей ВС), изменения условий окружающей среды, минимизации ошибок разработчиков, программистов, операторов. Улучшению надежностных характеристик отдельных компонентов и всей системы в целом способствуют: входной контроль компонентов ВС, повышение степени интеграции элементов (как следствие уменьшения паек и других

эффективные методы рассеива-
элементов. Однако данный подход
ограничения технического и эко-

устойчивых ВС допускает возникно-
при этом используются эффективные
последствий.

процессы возникновения и устране-
отказов и сбоев, а также способы восста-
вычислительного процесса. Процесс восстанов-
общего случая показан на рис. 6. Он может
простой вид, поскольку иногда отдельные
системы объединяются.



Последовательность состояний ОУВС

2.2.2. Классификация ошибок в работе ОУВС

В настоящее время используют следующую классификацию ошибок, возникающих в ОУВС [3, 14]:

1. По времени действия ошибок:

— постоянные—ошибки, связанные с отказом компонента ВС;

— временные (перемежающиеся) — ошибки, возникающие в результате неустойчивой работы компонента ВС;

— случайные — ошибки (сбои), связанные со случайными воздействиями окружающей среды на работу ВС.

2. По количественному признаку:

— одиночные — ошибки, вызванные сбоем или отказом одного компонента;

— множественные — ошибки, вызванные сбоем или отказом одного или нескольких компонентов и проявляющиеся в нескольких областях системы одновременно.

3. По моменту возникновения ошибок:

— ошибки, возникающие в период работы ВС (после восстановления);

— ошибки, возникающие во время восстановления ВС (повторные).

4. По характеру проявления ошибок:

— ошибка обращения — ошибка, вызванная обращением к запрещенному или несуществующему состоянию системы;

— ошибка сравнения — неправильный результат сравнения двух или более величин;

— ошибка задержки — совпадение времени выполнения идентичных операций;

— программная ошибка — неправильная работа программы (например, зацикливание).

Отказы компонентов происходят в основном в результате их «старения». Увеличение интенсивности отказов может быть связано с изменением параметров окружающей среды (например, повышением температуры, радиации и усилением вибраций).

2.2.3. Способы и средства контроля в ОУВС

Существует много разнообразных способов контроля работы ОУВС с целью обнаружения ошибок, их основ-

является использование избыточ-
[8, 26];
ность;
программного и аппаратного обеспе-
ность реализуется программными
некоторых дополнительных вычис-
для сравнения результатов отдельных
Их совпадение является осно-
работы системы.
ности предусматривает дополни-
и аппаратные средства, необходи-
контрольных операций и сравнения
существующих средств и способов вы-
структур ВС представлена на рис.7.
основные уровни выявления ошибок:
уровень. Средства, применяемые на этом
виде схем с самоконтролем, кото-
обнаружить ошибку сразу же, как только
Сложность полного «покрытия» всех ком-
контролирующими схемами связана с необ-
использования значительного количества из-
оборудования, а также с физической зависимо-
основными функциональными блоками и
выявления ошибок. Вследствие этой зависимо-
контролируемого компонента влечет за со-
средств выявления ошибок.
выявления ошибок на данном уровне можно
на следующие группы:
кодовые схемы;
с дублированием;
с голосованием (мажоритарные схемы).
происходит почти полное выявле-
можно считать маловероятный случай, когда
результата совпадают. В связи с успехами
аппаратуры этот способ представ-
перспективным. Например, в системе
троекратного дублирования отказав-
обнаруживается по несовпадению выходных

Рис. 7. Классификация способов и средств выявления и локализации ошибок

1. **Тестовый уровень.** Средства тестирования должны генерировать такие наборы входных сигналов (тестов), при которых все возможные отказы компонентов ВС вызывали бы ошибочные выходные сигналы, обнаруживаемые путем сравнения с правильными (эталонными) сигналами. Средства тестового уровня позволяют выявить постоянные ошибки и отчасти временные, случайные же ошибки ими не обнаруживаются.

Способы и средства тестирования делятся следующим образом:

По времени проведения тестирования:

В ходе работы ВС (при этом тестирование выполняется в неработающих подсистемах одновременно с основной работой других частей системы); в перерывах.

По периодичности проведения тестирования:

— периодическое — задача оптимизации периода контроля рассмотрена в работах [8, 14];

— непериодическое — (например, по окончании выполнения задания).

По продолжительности тестирования:

— в течение заданного промежутка времени;

— до локализации ошибки (в случае диагностического тестирования в предположении, что ошибка носит постоянный или временный характер, и при условии, что она была обнаружена средствами другого уровня).

Как показано в работах [5, 9], эффективность средств выявления ошибок есть монотонно возрастающая функция от времени тестирования.

По местонахождению средств тестирования:

— внутреннее тестирование (выполняется основными средствами процессорной системы), преимущества использования внутреннего тестирования для отказоустойчивых СБИС отмечены авторами в работе. Примером реализации такого тестирования могут служить системы System 4000, Segnola System. Подробный анализ мультипроцессорных систем такого рода дан в работе [17];

— внешнее тестирование (выполняется специальными — сервисными) процессорами системы.

По глубине тестирования, т.е. по точности определения местонахождения ошибки, средства тестирования отличаются количеством одновременно тестируемых элементов.

По способу организации тестирования:

— поочередное тестирование компонентов системы (обычно используется при внешнем тестировании ВС с большой степенью интеграции составляющих ее компонентов);

— одновременное (параллельное) тестирование компонентов ВС;

По способу проведения тестирования:

— самотестирование компонентов ВС;

— тестирование соседних (ближайших) компонентов ВС;

— тестирование любого процессорного элемента любым другим процессорным элементом ВС.

3. Функциональный уровень. Средства функционального уровня предназначены для предотвращения нежелательных действий системы, выявления ошибочной информации на уровне более высоком, чем сигнальный. Они представляют собой так называемые «барьеры» или «ограничения» вокруг верных результатов вычислений и правильных функциональных действий системы.

После отказа компонентов вызванная ими ошибка может очень быстро распространяться в среде ВС и «расти», создавая эффект «снежного кома» или ускорения ошибок, до падания в «барьер».

Средства данного уровня можно классифицировать по способу установления «барьера» в среде ВС следующим образом:

— временной «барьер» (контроль времени выполнения задания). Если задание не выполняется в критическое время, то задержка результата вычислений, которая является признаком ошибки, может привести к катастрофическим последствиям. Ошибку такого рода называют динамической ошибкой. Примерами реализации временного «барьера» могут служить такие системы, как Synapse N+1, Plus 32;

— пространственный «барьер» (контроль поля выполнения задания). Например, если данной программе выделено определенное поле памяти, то появление адреса, не принадлежащего этому полю, служит признаком ошибки. Ограничение распространения действия ошибок в одной области системы упрощает процесс восстановления ВС, так как при этом сохраняется большое количество информации, не подверженной действию ошибок, подтвержда-

ваются результаты вычислений. С помощью различного ряда дополнительных вычислений (посредством неосновного вычислительного процесса), либо подтверждаются результаты вычислений (основного процесса), либо в них выявляются ошибки. Здесь могут использоваться методы контрольных функций, контрольных сумм и др. [24].

Заметим, что для построения высоконадежных ОУВС недостаточно использования средств выявления ошибок какого-то одного из уровней. Только совокупное применение средств всех трех уровней может дать желаемый результат.

2.2.4. Способы и средства устранения ошибок и отказов ОУВС

Простейшим способом устранения ошибок является повторение вычислений. Однако он позволяет устранить только ошибки, вызванные сбоями и, кроме того, требует значительных затрат машинного времени. В практике встречаются два основных способа устранения последствий отказов и ошибок ОУВС (рис. 8):

— маскирование ошибочных действий;

— реконфигурация системы.

Суть первого способа состоит в том, что избыточная информация скрывает действие ошибочной информации благодаря особенностям схемных решений и организации вычислительного процесса. При этом используются статические средства устранения последствий ошибок — средства маскирования.

Средства маскирования можно разделить по принципу действия на следующие группы:

— корректирующие коды (коды Хэмминга и др.);

— логика с переплетениями;

— схемы с голосованием.

В последнем случае используется нечетное число блоков, выполняющих одни и те же вычислительные операции, и большинством «голосов» определяется правильный выбор выходных сигналов.

Второй способ — реконфигурация ВС — заключается в изменении состава вычислительных средств и (или) способа их взаимодействия (рис. 8). Реконфигурация произво-

Рис. 8. Классификация способов и средств устранения последствий ошибок и отказов

дится после выявления отказа. Этот способ устранения последствий ошибок и отказов включает:

— статическую реконфигурацию;

— динамическую реконфигурацию.

Статическая реконфигурация системы осуществляется путем отключения неисправных компонентов. При этом система делится на две части: активную — непосредственно участвующую в работе ВС, и пассивную — охватывающую неработоспособные компоненты системы и отключенные в ходе реконфигурации.

Динамическая реконфигурация по принципу проведения делится на следующие виды:

— замещение — поддержка запасом (примером построения ОУВС на основе этого принципа являются системы Tandem NonStop I, II [14, 17].

— дублирование (примером может служить система Sunapse 4000).

Постепенная «деградация» системы (снижение вычислительных способностей). По этому принципу построены такие системы, как Power 5/55.

Если система имеет маскирующую избыточность как часть схемы динамической реконфигурации, то удаление отказавших компонентов можно отложить до наступления следующего отказа, когда общее количество отказавших компонентов станет угрожающим в смысле возникновения немаскированной ошибки.

2.2.5. Способы восстановления ОУВС

После реконфигурации для продолжения нормальной работы системы необходимо ее восстановить, что происходит на двух уровнях (рис. 9).

1. Аппаратный уровень. Здесь производится восстановление отказавших компонентов ВС двумя способами [6, 13, 14].

Первый способ – восстановление, реализуемое путем дополнительной реконфигурации системы. При этом предполагается, что в системе имеется ряд запасных блоков,

Рис. 9. Классификация способов восстановления ОУВС

благодаря которым она возвращается в работоспособное состояние. Производительность системы либо сохраняется, либо может быть несколько снижена.

Ремонт (восстановление вручную). В этом случае отказавший блок отключается от системы, она либо продолжает работу с меньшей производительностью, либо приостанавливается до возвращения отремонтированного блока в активную часть ВС. Как правило, после ремонта производят повторный запуск всех программ. Примером такой системы является CPU Parallel Computer Inc.

2. Программный уровень. Здесь осуществляется восстановление информации о состоянии ВС, необходимой для продолжения ее работы. В зависимости от нарушений в работе системы (от количества ошибочной информации) можно выделить следующие способы восстановления:

— повторение операции или последнего действия на различных уровнях (команд или микрокоманд), повторное выполнение некоторых может дать правильный результат, если связанная с ними ошибка является случайной или временной (ошибка исчезает в процессе восстановления). Часто практикуется многократное повторение последнего действия или операции, что увеличивает вероятность восстановления правильности вычислительного процесса;

— возвращение к контрольной точке. Контрольной точкой называется некоторый этап вычислительного процесса, для которого зафиксированы (в одном или нескольких ЗУ) промежуточные результаты вычислений и информация о состоянии системы, позволяющая возобновить вычисления. Возвращение к контрольной точке можно рассматривать как разновидность общего случая повторения вычислительного процесса. При обнаружении ошибки система возвращается к контрольной точке, предшествующей моменту возникновения отказа, и продолжая свою работу, используют данную точку в качестве исходной (при условии, что все результаты в данной точке программы безошибочны и сохранены);

— повторное выполнение программы. При этом способе восстановления все неоконченные (до возникновения отказа) программы выполняются с самого начала. Это необходимо, когда в системе разрушено такое количество

информации, что восстановление путем повторного выполнения отдельных операций или участков программ невозможно или не имеет смысла. Данный способ применяется в трех случаях:

1) если последствия ошибочных действий успели отразиться на большей части системы;

2) если существует возможность восстановления только части вычислительных процессов, даже при минимальном количестве имеющихся в них ошибок;

3) если продолжение работы системы при использовании других способов восстановления сопряжено с большими трудностями или большими временными затратами.

Принцип повторного выполнения программ реализован в системе CPU Parallel Computer Inc.

Каждый из рассмотренных способов восстановления вызывает задержку в выполнении задания. Использование средств восстановления программного уровня — повторного выполнения программ, фрагментов программ или отдельных операций — требует больших временных затрат, чем маскирование ошибок. Однако определить оптимальное сочетание средств маскирования и восстановления сложно, тем более, что допустимая задержка в восстановлении зависит от конкретной области применения ОУВС.

Прогрессивное развитие элементной базы средств вычислительной техники позволяет уже сегодня включить различные средства обнаружения и исправления ошибок, а также восстановления ВС в состав отказоустойчивых вычислительных систем. Примерами построения таких систем могут служить IBM 4300 и Univas 1100/60.

В заключение можно сделать следующие выводы:

— наиболее целесообразно использовать сочетание различных средств обнаружения и локализации ошибок и восстановления ОУВС, так как каждое из них с высокой степенью достоверности обнаруживает лишь определенный класс ошибок;

— применение различных средств контроля и восстановления наиболее эффективно, если оно ведется на различных иерархических уровнях системы;

— наиболее эффективным и перспективным способом контроля, используемым при построении ОУВС, является контроль дублированием.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Укажите общие черты, присущие всем известным ОУВС.
2. Как обеспечить высокую надежность при создании вычислительных систем?
3. Укажите ошибки, возникающие в ОУВС по количественному признаку.
4. Определите основные уровни выявления ошибок в ОУВС.
5. В каких случаях применение средств контроля и диагностирования в ОУВС наиболее эффективно?
6. Какую ошибку в ОУВС легче выявить: жесткую или мягкую?
7. Дайте определение понятию «реконфигурация».
8. В чем суть динамической реконфигурации? На какие виды она делится по принципу проведения?
9. Как в ОУВС реализуется автоматическое восстановление?
10. Какой вид контроля в ОУВС является наиболее перспективным?

Глава 3. МЕТОДЫ ПРОЕКТИРОВАНИЯ ЛОКАЛЬНЫХ СРЕДСТВ ОБЕСПЕЧЕНИЯ ОТКАЗОУСТОЙЧИВОСТИ

3.1. СРЕДСТВА ОБНАРУЖЕНИЯ ОШИБОК

Обнаружение факта возникновения ошибки — это начальная точка реализации всех остальных функций СОО (за исключением такого способа реализации СОО, как маскирование неисправностей, в частности мажоритарное резервирование). Самые сложные методы восстановления информации и аппаратуры дают положительный эффект лишь постольку, поскольку эффективны схемы обнаружения ошибок в работе элементов ВС.

На рис. 10 показана последовательность наступления случайных событий при возникновении отказа в системе, где видно, во-первых, что если отказ не обнаружится, то на неопределенное время откладывается процесс восстановления и информации и аппаратуры, а пользователь может получить неверный результат решения его задачи, во-вторых, чем больше время t_2, тем больше искажений вносится как в процесс исполнения программы, так и в обрабатываемые данные.

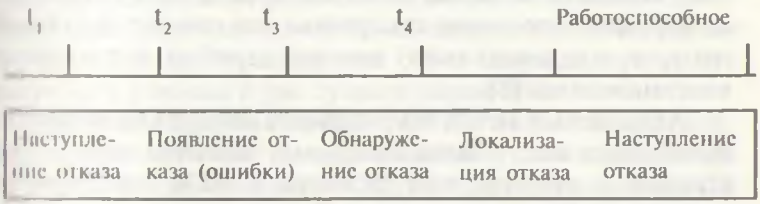


Рис. 10. Последовательность наступления случайных событий при возникновении отказа в системе

Кроме того, при построении схем обнаружения ошибок нужно учитывать возможность ложного сигнала ошибки, вызванного сбоем или отказом самой схемы обнаружения.

Следовательно, разработка системы обнаружения ошибок является основополагающей при построении средств обеспечения отказоустойчивости ВС.

Обнаружение ошибок и, следовательно, отказов, которые эти ошибки вызывают, может быть реализовано как аппаратным, программным и временными методами, так и комбинацией их. С точки зрения реализации процесса обнаружения ошибок обычно используется термин «контроль».

Контроль может быть начальным, оперативным и плановым. *Начальный* контроль осуществляется перед нормальной эксплуатацией и служит для выявления неисправных элементов аппаратуры, дефекты которых возникли в ходе производства или сборки. *Оперативный* контроль осуществляется одновременно с нормальной работой системы и служит для выявления вновь возникших отказов в минимально возможный промежуток времени. *Плановый* контроль осуществляется при заранее предусмотренном временном перерыве работы ВС.

Наиболее сложной является организация оперативного контроля (ОК). Обнаружение ошибок с помощью ОК может быть реализовано посредством введения в систему специальных аппаратных или программных средств, работающих параллельно с основными аппаратно-программными средствами ВС. Важным преимуществом оперативного обнаружения ошибок является то, что сигнал ошибки в системе вырабатывается до того, как последствия отказа резко нарушат исполнение программы или нанесут ущерб данным, и, следовательно, можно перейти к процедуре восстановления [8].

Аппаратный метод оперативного обнаружения ошибок применялся еще в вычислительных машинах первого поколения и получил свое развитие в ЭВМ последующих поколений. К нему относится использование кодов с обнаружением ошибок (контроль по модулю и т.д.), дублирование со сравнением результатов, применение детекторов расхождения, с мажоритарными схемами голосования, специальных схем контроля некоторых ответственных элементов (генераторов тактовых импульсов, источников питания, схем записи в память и т.д.).

Программный метод оперативного обнаружения ошибок основан на параллельном исполнении программ или же сводится к введению в программу каких-то дополнительных особенностей. В первом случае работают несколько одинаковых программ, пользующихся отдельными процессорами и отдельными запоминающими устройствами и, следовательно, необходимы соответствующие избыточные аппаратные компоненты ВС. Сравнение результатов осуществляется не с помощью схем, а посредством программного обмена результатами или контрольными суммами. Возможно также использование специализированной подсистемы (например, в виде контролирующей ЭВМ), исполняющей контрольные программы с целью наблюдения за работой остальной части системы.

Следует отметить, что несмотря на то, что данный метод называется программным, в системе должны присутствовать дополнительные аппаратные средства. Во втором случае средства обнаружения ошибок, которые могут быть введены в вашую программу, включают в себя пароли подтверждения, контрольное суммирование, проверку на выполнение определенных контрольных соотношений при вычислении промежуточных результатов (например, $\sin^2 x + \cos^2 x = 1$), проверку результатов на общее соответствие, (например, значение вероятности какого-либо события не может быть больше 1 или меньше 0), программированные контрольные таймеры и т.д. По сравнению с аппаратными реализациями обнаружения ошибок программные работают медленнее и больше подвержены опасности нарушения из-за самих отказов элементов системы. Но они используются очень широко, потому что могут быть введены в уже существующие ВС.

Наиболее перспективным в ОУВС является организация аппаратного ОК с использованием самопроверяемых схем встроенного контроля (ССВК). Такая организация ОК позволяет выявить ошибки, вызванные не только отказами основных элементов системы, но и отказами самих СОО.

В отличие от традиционных схем встроенного контроля, которые вырабатывают единичный сигнал ошибки на основе, например, контроля по модулю, ССВК (на основе того же контроля по модулю) должны вырабатывать как минимум два сигнала ошибки. В работе [15] дается такое

нормальное определение ССВК. Схема встроенного контроля с двумя выходами — f_1 и f_2, является полностью самопроверяемой, если она обладает следующими двумя свойствами:

а) самотестируемость — все неисправности ССВК из заданного класса проявляются на выходах f_1 и f_2 в виде пар значений 00 или 11, хотя бы на одном входном наборе из множества установленных наборов при исправном контролируемом устройстве;

б) защищенность от неисправностей — каждая неисправность ССВК из заданного класса проявляется на выходах f_1 и f_2 только в виде пары значений 00 или 11.

При произвольной реализации схема встроенного контроля, имеющая два выхода, — f_1 и f_2, будет гарантированно обнаруживать неисправности только контролируемого комбинационного устройства, нарушающие его правильное функционирование. Поэтому одним из условий построения ССВК является требование раздельной реализации функций f_1 и f_2. Обоснование этого требования состоит в следующем: неисправность ССВК будет обнаружена лишь при изменении значений выходов f_1 и f_2 с 01 или 10 на 00 или 11, т.е. при инвертировании значения только одного выхода. Раздельная реализация функций f_1 и f_2 гарантирует, что любой одиночный отказ на входах и выходах элементов ССВК не приведет к инвертированию обоих выходов схемы. Таким образом, при раздельной реализации функций f_1 и f_2 заведомо выполняется свойство защищенности от неисправностей. Рассмотрим пример построения схем контроля дешифратора.

На рис. 11 приведена схема дешифратора (ДС) и схема его контроля М2 — схема сложения по модулю — два значения выходов ДС. Принцип контроля ДС основан на том, что при нормальной его работе только один выход имеет значение логической единицы — «1», остальные имеют значение логического нуля — «0». Поэтому на выходе схемы М2 всегда должна быть «1». В этом случае ошибка на выходах дешифратора, которая приводит к четному числу единиц на выходах или пропаданию единицы на выбранном выходе, всегда будет обнаружена, так как на выходе М2 появится «0». Данная схема обладает тем недостатком,

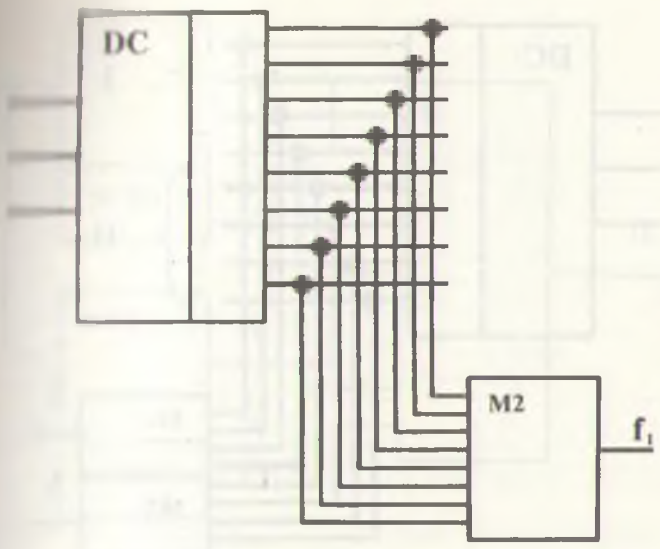


Рис. 11. Схема дешифратора (ДС) и схема его контроля

что отказ схемы М2, который приводит к константной ошибке типа «1», на выходе не будет обнаружен, а отказ, который приводит к ошибке типа «0», будет обнаружен, но при этом нельзя установить, отказал ли это ДС или М2.

На рис. 12 представлена схема ДС ССВК, которая имеет два выхода f_1 и f_2, при этом организация ее простая за счет отмеченной выше особенности ДС.

При правильной работе ДС и М2 выходы f_1 и f_2 принимают значения 01 или 10. При пропадании 1 на выходе ДС значения выходов f_1 и f_2 становятся 00; при нечетном числе 1 на выходах ДС значения выходов f_1 и f_2 принимают 00 и 11. Следовательно, такие ошибки могут быть обнаружены. Кроме того, если откажет одна из схем М2, при этом на ее выходе будет либо константная «1», либо константный «0», то всегда существует при правильной работе ДС такая комбинация значений на его входах, которая приведет к значениям f_1 и f_2, либо 11, либо 00 соответственно. Следовательно, любая ошибка схем контроля обнаруживается.

Для того чтобы различить, какая именно схема отказа-

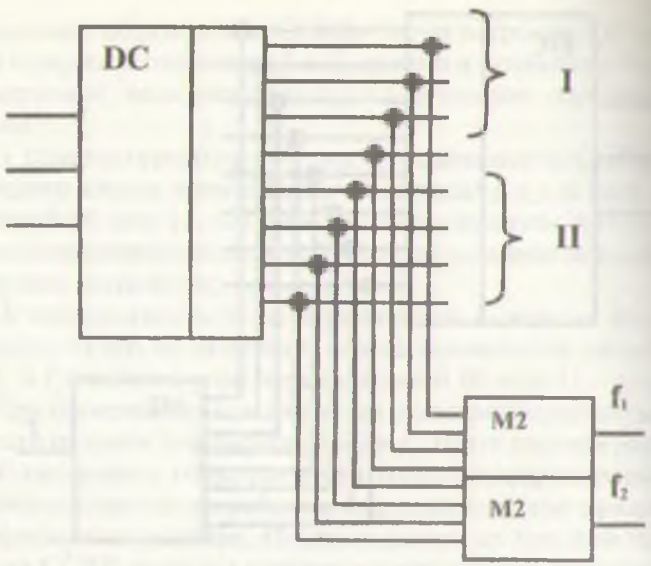


Рис. 12. Схема дешифратора ССВК с двумя выходами

ла, необходима более сложная организация контроля, которая приводит к дублированию схем контролируемых и контролирующих. На рис. 13 представлена схема дублирования операционного блока (ОБ), обладающего самопроверяемыми схемами встроенного контроля. Схемы ССВК позволяют обнаруживать ошибки заданного типа, как в ОБ, так и в ССВК. Схема коммутации (К), которая является мультиплексором, выдает на выход результат либо с ОБ1, либо с ОБ2 в зависимости от сигналов f_1^1 и f_2^2 и f и f_2^{11}.

Таким образом, подобная организация средств контроля позволяет не только обнаруживать ошибку, но и за счет дублирования устранять ее влияние на правильность выходного результата при условии, что ошибка произошла только в одном из дублированных блоков. В результате одновременно реализуются две функции СОО — обнаружение ошибки и восстановление информации, при этом практически без снижения производительности основной аппаратуры.

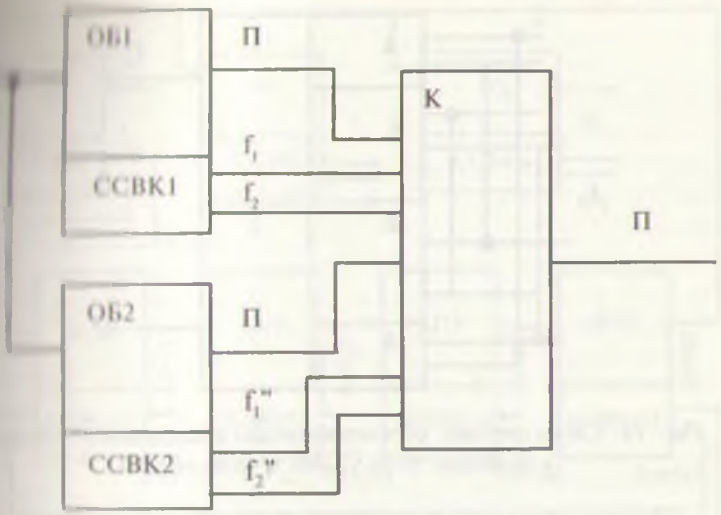


Рис. 13. Схема дублирования операционного блока с самопроверяемыми схемами встроенного контроля

Обычно в состав ОУВС входит большое число устройств, каждое из которых в свою очередь состоит из различных функциональных узлов. При этом все функциональные узлы должны быть оснащены ССВК, которые вырабатывают сигналы ошибок f_1 и f_2. Но для того, чтобы запустить в системе алгоритмы функций восстановления, необходимо выработать общий сигнал ошибки. Для объединения сигналов с выходов ССВК с целью формирования общего сигнала ошибки применяют самопроверяемые схемы сжатия. Схема сжатия, приведенная на рис. 14, обеспечивает объединение сигналов с выходов двух ССВК в один общий. Если на входах схемы сжатия сигналы f_{11} и f_{12} или f_{21} и f_{22} принимают значения 00 или 11, ошибка произойдет в самой схеме сжатия и выходы f_{31} и f_{32} примут значения 00 или 11, т.е. будут свидетельствовать о наличии ошибки. Следует также иметь в виду, что формируя общий сигнал ошибки в системе (устройстве), необходимо сохранить и информацию о первоначальном источнике ошибки. Это позволит значительно эффективнее реализовать все функции группы восстановления.

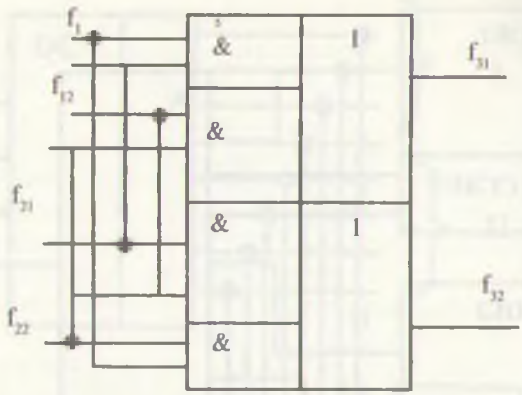


Рис. 14. Схема сжатия, обеспечивающая объединение сигналов с выходов двух ССВК в один общий

Таким образом, если архитектура ОУВС предусматривает введение аппаратного оперативного контроля с помощью ССВК, структурного резервирования и аппаратной реализации большинства функций восстановления (обычно функция распознавания сбоев или отказов аппаратуры системы реализуется программно — повтором того или иного участка программы), то примером структуры организации СОО может служить структура, приведенная на рис. 15.

В состав ОУВС входит n модулей M_i (устройств или сложных функциональных узлов), предназначенных для выполнения функций, возложенных на них как на элементы вычислительной системы. При этом в состав n модулей могут входить как основные, так и резервные. В каждый из n модулей введены ССВК, которые вырабатывают сигналы ошибок (возможно с использованием схем сжатия) и направляют их в автономную аппаратуру СОО. Эта аппаратура состоит из блока БСИ — сбора и хранения информации об ошибках в системе и выработке сигнала прерывания о наличии ошибки (СПО), блока БАИ — анализа информации об ошибках и состояниях элементов системы, блока БПР – принятия решения по изменению алгоритма функционирования системы, блока БПС — исполнения перестройки системы. Каждый из указанных блоков должен быть снабжен схемами контроля типа ССВК.

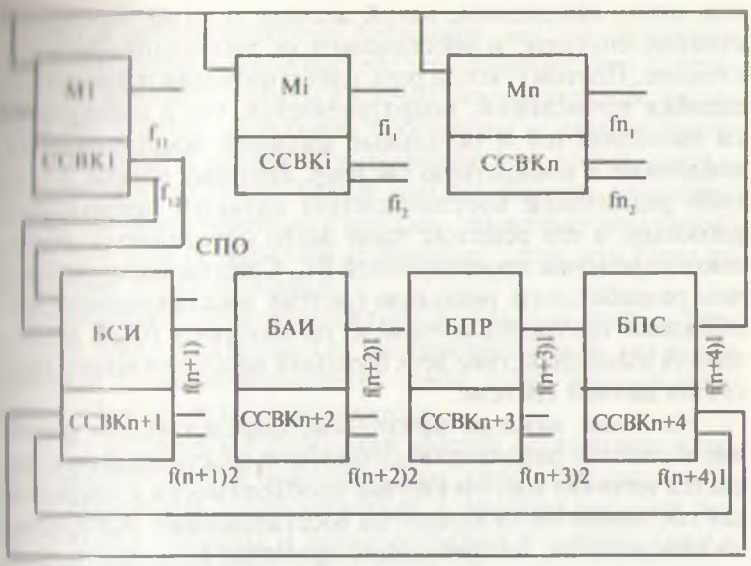


Рис. 15. Структура организации СОО

3.2. СРЕДСТВА ВОССТАНОВЛЕНИЯ РАБОТОСПОСОБНОГО СОСТОЯНИЯ

Как было отмечено выше, система восстановления при автономном способе ее реализации начинает действовать лишь тогда, когда появился сигнал, фиксирующий ошибку в вычислениях в том или ином узле ВС. В этом есть некоторое противоречие. С одной стороны, мы создаем сложную систему аппаратно-программных средств, реализующих функции восстановления, работа которых должна отличаться высокой надежностью и которые должны включаться в работу в любой момент времени, а с другой стороны, мы создаем ВС, в которой сбои и отказы аппаратуры должны возникать как можно реже, и, следовательно, эти средства должны включаться в работу относительно редко.

Обнаружив ошибку, можно исправить ее последствия (т.е. восстановить искаженную ею информацию) и возобновить правильно вычислительный процесс только в том случае, если установить, был ли причиной ошибки сбой

или отказ, определить, какой элемент отказал, каково состояние системы, и восстановить ее работоспособное состояние. Поэтому, когда речь идет о функции исправления ошибки вычислений, подразумевается, что в совокупности выполняются и остальные функции восстановления, введенные в конкретную систему. Поэтому вопрос о способе реализации восстановления является чрезвычайно сложным, а его решение чаще всего определяется областью применения проектируемой ВС. Следовательно, прежде чем разрабатывать реальную систему восстановления, необходимо построить алгоритм, по которому будет происходить взаимодействие всех функций восстановления, присущих данной системе.

Наиболее важным признаком, определяющим различие в способе реализации алгоритмов восстановления, является наличие или отсутствие необходимости в операторе как составной части алгоритма восстановления. Алгоритмы восстановления, не требующие принятия решения человеком, являются автоматическими, остальные — ручными, хотя в состав последних могут входить и автоматические (программные) модули. Алгоритм автоматического восстановления может также предусматривать последующий ремонт отказавшего сменного блока, выполняемый вручную. Существенно, однако, чтобы возобновление нормальной работы в этом случае не зависело от вмешательства человека. В соответствии с состоянием системы, в которое она переходит после завершения восстановления, автоматические алгоритмы восстановления делятся на три класса: полное восстановление, частичное и безопасный останов.

Полное восстановление означает, что система вернулась (в пределах установленного срока) к тем условиям работы, которые существовали до возникновения неисправности. Как аппаратное, так и программное обеспечение при этом сохраняют прежнюю вычислительную мощность, отказавшие модули аппаратуры заменяются на запасные, а поврежденная информация (программы и данные) возвращаются в нормальное, известное состояние, имевшее место до появления неисправности.

Частичное восстановление (системы, в которых оно реализуется, называются деградирующими) возвращает

систему в работоспособное состояние, но при снижении вычислительных возможностей. Это означает, что отдельные отказавшие элементы системы исключаются из ее состава без замены, отдельные программы и данные утеряны, а некоторые функции выполняются за время, превышающее допустимое. Такой подход иногда называют частичной устойчивостью к отказам, поскольку восстановление при нем неполно по сравнению с тем, что было до появления неисправностей.

Безопасный останов является предельным случаем частичного восстановления. Он выполняется тогда, когда оставшаяся производительность упала ниже приемлемого порога. Целями останова являются:

1) не допустить причинения ущерба оставшейся в памяти информации и поврежденным элементам системы;

2) прекратить взаимодействие с другими частями более сложной системы и пользователями в соответствии с заранее установленным порядком;

3) передать сообщение об останове и диагностическую информацию предписанным системам, пользователям или специалистам по техническому обслуживанию.

В ОУВС обычно реализуется автоматический алгоритм восстановления, при этом в ряде случаев эффективнее использовать последовательность из полного, частичного восстановления и безопасного останова. Первое выполняется до тех пор, пока не задействованы все элементы структурного резервирования, затем выполняется частичное восстановление. Если возможно восстановление отказавших элементов и возвращение их в рабочую конфигурацию системы, то вновь может действовать полное восстановление. В случае, если полностью исчерпан и структурный, и функциональный резерв, начинает действовать безопасный останов.

Как было отмечено выше, реализация СОО в сильной степени определяется типами отказов, устойчивость к которым необходимо обеспечить. При реализации алгоритмов восстановления следует различать отказы по степени воздействия их на вычислительный процесс в ВС. С этих позиций различают мягкие и жесткие ошибки [5]. Мягкими называют ошибки, вызванные чаще всего сбоями, по-

следствия которых удается автоматически устранить, и ис
становить выполнение программы без вмешательства опе
ратора. Жесткими называют ошибки, если СОО не в состо
янии автоматически восстановить выполнение программы
и для восстановления вычислительного процесса может
потребоваться реконфигурация системы с ручным отклю
чением отказавшего блока, перегрузка операционной сис
темы или другие действия обслуживающего персонала.

Следовательно, создать систему восстановления необ
ходимо так, чтобы возможно большая часть ошибок про
являлась как мягкие, а это в свою очередь, позволяет ус
тойчиво работать системе с заданной производительностью.

3.3. СРЕДСТВА ВОССТАНОВЛЕНИЯ ИНФОРМАЦИИ

Система восстановления ОУВС может быть построена либо на основе централизованного управления, либо децентрализованного. В первом случае необходим автономный орган, который на основе поступившей информации об ошибках в элементах систем выполняет процедуры управления восстановлением искаженной информации и отказавшей аппаратуры. Примером такой системы служит система Star, которая будет описана ниже. Во втором случае исправление мягких ошибок выполняется локальными средствами восстановления, имеющимися в элементах системы. Такая организация системы восстановления возможна при применении либо мажоритарного резервирования, либо в том случае, когда элементом системы является процессор или ЭВМ, способные, например, самостоятельно восстановить вычислительный процесс, нарушенный сбоем аппаратуры, повтором выполнения участка программы. При этом следует иметь в виду, что в последнем случае структура локальных средств восстановления должна содержать блок сбора информации об ошибках, блок классификации ошибок (сбой или отказ), блок автоматической реконфигурации, блок выработки сообщений о состоянии элемента в центральный орган управления системой или оператору.

Рассмотрим алгоритм восстановления вычислительного процесса после возникновения мягкой ошибки с помощью локальных средств восстановления ЭВМ. В общем виде

[illegible] можно представить как последовательность следующих действий. При возникновении сигнала ошибки от схем контроля необходимо распознать, вызвана ли она сбоем или отказом аппаратуры. Если произошел сбой, то восстановление осуществить по одной соответствующей процедуре, если же произошел устойчивый отказ, то последний надо классифицировать на допускающий и не допускающий восстановление путем автоматической реконфигурации. Когда реконфигурация невозможна (исчерпаны все ресурсы — резервные элементы), то система восстановления должна выработать сообщение о состоянии элементов. Классификация ошибки на сбой или отказ производится путем многократного повторения участка программы, команды или микрокоманды, при выполнении которого произошла ошибка. Если одно из повторений ошибочной операции завершается правильно, ошибка классифицируется как сбой, производится регистрация информации о сбое и соответствующем состоянии ЭВМ, и восстанавливается выполнение программы. Когда при повторениях не удается выполнить операцию правильно, ошибка классифицируется как отказ, и тогда необходимо произвести локализацию отказа. После регистрации данных о состоянии ЭВМ на основе этих данных принимается решение о возможности реконфигурации. При положительном решении производится реконфигурации и восстановление вычислительного процесса.

На рис. 16 представлена блок-схема алгоритма восстановления, которая отражает последовательность действий при выполнении программы П и возникновении сигнала прерывания от схем контроля при обнаружении ошибки на участке программы между контрольной точкой 1 (КТ1) и контрольной точкой 2 (КТ2) .

Любая вычислительная система состоит из сложных элементов, различающихся функциональным назначением, которое требует специфического способа организации обнаружения ошибок и их исправления. В соответствии с этим различают аппаратуру памяти, процессора, канала и периферийных устройств. Для восстановления информации после сбоя или устойчивого отказа элемента запоминающей среды, применяют аппаратный метод, который связан с применением корректирующих кодов. Чаще всего

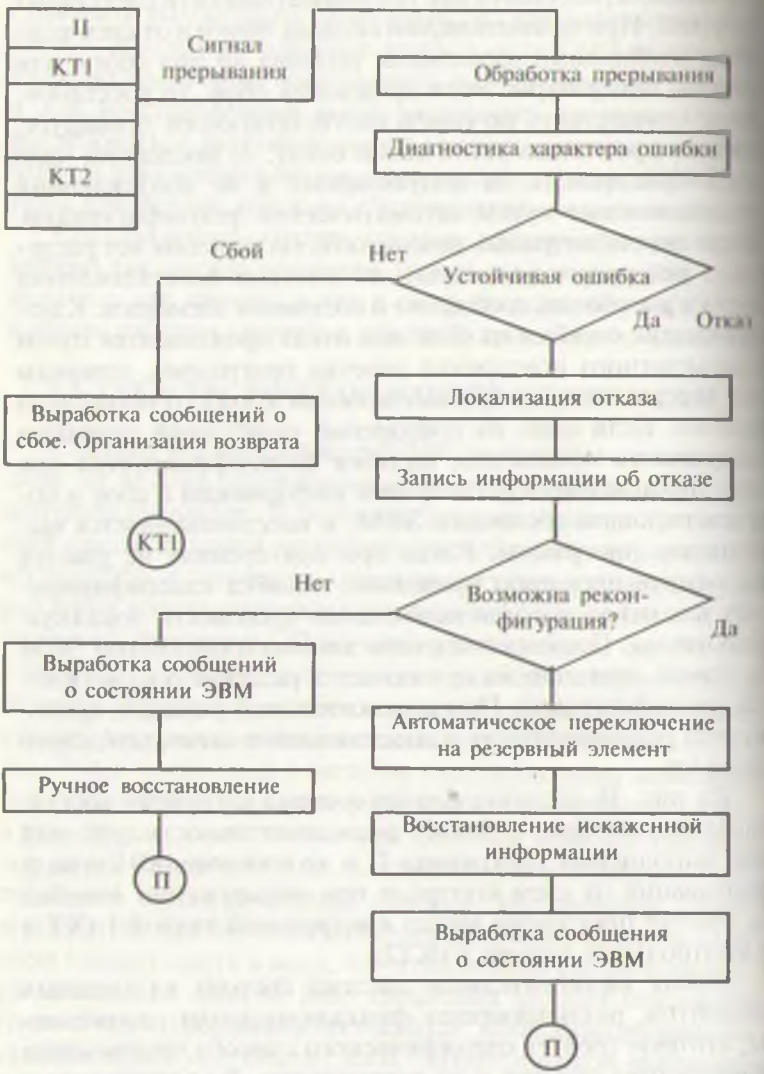


Рис. 16. Аппаратно-программные методы восстановления после сбоев процессора (организация повтора микрокоманды, команды или участка программы)

ется код Хэмминга, позволяющий исправлять одиночные ошибки (иногда и двойные) в слове. В этом случае сигнал прерывания от схем контроля не вырабатывается, исправление ошибки производится специальными схемами коррекции, и вычислительный процесс продолжается нормальным порядком. Для восстановления после сбоев процессора чаще всего используются аппаратно-программные методы. Эти методы заключаются в организации повтора микрокоманды, команды или участка программы, как это показано на рис. 16. Для организации повтора микрокоманды, выполняющейся в процессоре в момент сбоя, необходимы: сохранение информации о данной микрокоманде, устранение последствий сбоя (для того, чтобы в дальнейшем они не вызывали других ошибок), восстановление состояний определенных индикаторов, имевших место, наличие исходных данных для выполнявшейся операции, организация повторной попытки выполнения операции.

Для организации повтора команды необходимы условия, которые требуются для повтора микрокоманды. Однако, в зависимости от способа организации выполнения команд в процессоре возможно введение дополнительных регистров для сохранения исходных данных (операндов команды). Применяется в ЭВМ и метод контрольных точек. Этот метод связан с разбиением программы на участки, ограниченные контрольными точками.

При выполнении участка программы в процессоре сохраняется вся информация о текущем состоянии вычислительного процесса на начало данного участка, т.е. контрольной точки. Поэтому после сбоя при выполнении какой-либо операции участка программы необходимо восстановить эту информацию и начать повторное выполнение участка программы.

Следует иметь в виду, что возможны такие ошибки в процессоре, которые не позволяют автоматически перейти к повтору микрокоманды, команды или участка программы. Тогда в случае применения данного метода восстановления необходимы дополнительные меры в рамках ОУВС. В качестве возможной меры можно применить так называемый временной контроль, который заключается в

том, что некоторый центральный орган ОУВС или другой процессор должен регулярно получать информацию о состоянии рассматриваемого процессора (особенно в том случае, если ошибка системой контроля обнаружена). Если такая информация в определенный заранее момент не поступила, принимается решение о диагностировании данного процессора и возможной замене его на резервный.

Восстановление информации из-за сбоя в канале происходит либо с использованием корректирующих кодов, либо повтором команды обмена по сигналу ошибки от схем контроля передачи информации. При этом организация повтора команды определяется применяемым в системе интерфейсом.

Восстановление от сбоев периферийных устройств (ПУ) обычно выполняется с помощью программных средств. Для этого в операционных системах для всех типов ПУ имеются программы обработки ошибок, данные программы анализируют тип ошибки, при сбое устройства уточняют состояние ПУ и пытаются исправить сбойную ситуацию путем многократного повторения начальной программы, при выполнении которой произошел сбой. Число повторений зависит от типа ошибки и ПУ.

Таким образом, данные способы позволяют восстанавливать искаженную информацию, вызванную в основном сбоями, и применяются в системах, допускающих задержку выдачи результата обработки на время повтора соответствующей части программы. Исправление ошибок возможно либо с применением корректирующих кодов, либо за счет структурного резервирования.

3.4. КОМПЛЕКСНЫЕ СРЕДСТВА ВОССТАНОВЛЕНИЯ

Как было отмечено выше, существуют способы организации СОО, позволяющие совместить реализацию функций обнаружения ошибок, их локализации и исправления. К этим способам относятся такие виды постоянного резервирования, как мажоритарное со схемами определения отказавшего канала и параллельное со схемами ССВК. Основной особенностью указанных видов резервирования является то, что неисправности как бы маскируются избы-

[illegible] аппаратурой и тем самым для программного обеспечения остаются невидимыми, т.е. эффект неисправности (ошибки) не выходит за рамки соответствующего модуля системы. Пока избыточность не исчерпана, неисправность остается скрытой внутри модуля и никак не проявится на его выходах. Если же избыточность полностью исчерпана или не в состоянии справиться с возникшей неисправностью, происходит отказ модуля. Рассматривая модуль извне, невозможно выделить отдельно функцию восстановления.

Главным вопросом при разработке системы маскирования является выбор размеров модуля, внутри которого осуществляется маскирование, в качестве модуля может быть выбран дискретный компонент системы — реле, микросхема, может — процессор или ЭВМ, а возможно — вся вычислительная система. Определение рационального размера модуля возможно только с помощью теоретического анализа на основе требований технического задания к системе, интенсивностей отказов компонентов и возможностей реализации маскирования на различных уровнях детализации. Например, трудно обосновать применение маскирования внутри интегральных схем, где многие неисправности способны повлиять на целый ряд соседних компонентов, что приводит к возникновению множественной ошибки, неисправляемой избыточными схемами. Это связано с тем, что маскирование основано на предположении о том, что отказы элементов взаимно независимы.

С помощью маскирования можно исправлять ошибки, вызванные как сбоем, так и постоянными отказами элементов. При этом способе организации постоянного резервирования не различаются элементы основные и резервные. Все элементы постоянно включены в схему и, постоянно получая питание, одновременно выполняют заданные модулю функции. Благодаря этому практически мгновенно и автоматически обеспечивается маскирование неисправностей. Однако, если избыточность исчерпана или если возникший отказ не поддается маскированию, то на выходе модуля появляется ошибочный результат, при этом восстановление информации в пределах модуля, даже с задержкой, не обеспечивается.

К другим недостаткам маскирования относится сто-

имость многократного резервирования, в три и более раз превышающая стоимость неизбыточной исходной системы. Однако данные способы организации СОО используются в системах благодаря простоте реализации и тому, что осуществляют мгновенное исправление ошибок. Следует добавить, что на примере таких видов резервирования отчетливо виден сам принцип свойства отказоустойчивости.

Рассмотрим примеры реализации комплексных средств восстановления. На рис.17 представлена структурная схема безызбыточного элемента системы, состоящего из 4 модулей. Для данного элемента на основе предварительного анализа решено применить такой вид постоянного резервирования, как помодульное мажорирование 2 из 3 со схемами определения отказавшего канала. Такое резервирование позволяет маскировать любую одиночную ошибку, возникшую в каждом модуле системы.

Само маскирование, т.е. исправление ошибки, происходит в специальных схемах, называемых восстанавливающими органами (ВО). На рис.18 показана функциональная схема ВО, применяемого при мажорировании модуля, имеющего один выход. На рис. 19 представлена структурная схема избыточного элемента системы, в котором каждый исходный модуль представлен группой из трех одинаковых, работающих в идентичных режимах. В каждый момент времени при правильной работе всех элементов схемы на входы модулей каждой группы поступает одна и та же информация, и, следовательно, на их выходах также должен быть одинаковый результат.

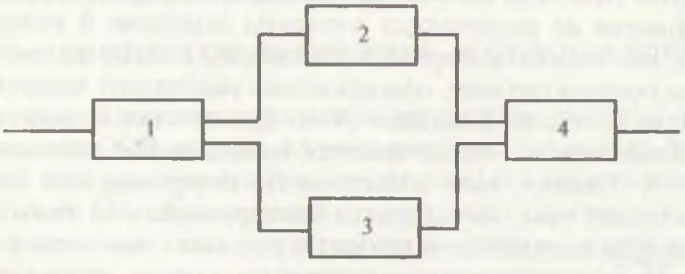


Рис. 17. Структурная схема безызбыточного элемента системы, состоящего из 4-х модулей

Рис. 18. Функциональная схема [illegible] применяемого при мажоритарном модули, имеющего один выход

Если в одном из модулей группы возникнет ошибка, то она исправится ВО, принадлежащим этой группе. Сами ВО также представляют собой избыточную группу из трех схем. Это необходимо для того, чтобы исправлять ошибки, возникающие в самих ВО. Но эти ошибки будут исправлены только ВО, следующими за группой модулей, на которые поступает искаженная информация. Например, если ошибка произошла в ВО1, то она исправится в восстанавливающих органах ВО2. При этом модули 2′ и 2″ должны быть исправны. На рис.19 показаны также схемы определения отказавшего канала М2. Эти схемы, выполняющие функцию сложения по модулю 2, производят сравнение сигнала, поступающего на вход ВО с выхода соответствующего модуля, с сигналом, вырабатываемым ВО. При несовпадении данных сигналов вырабатывается сигнал ошибки (СО), свидетельствующий о том, что произошла ошибка либо в соответствующем модуле элемента системы, либо в ВО.

Таким образом, данный способ реализации резервирования позволяет обнаружить ошибку, исправить ее и указать место ее возникновения с точностью до модуля и ВО практически в момент ее возникновения. Следует отметить, что определить, был ли это сбой или отказ, повлекший ошибку, можно только с помощью внешних по отношению к данному, элементов системы за счет повтора выполнения операций и анализа его результатов. Как видно из рис. 19, на этом примере отчетливо проявляется основной недостаток мажоритарного резервирования — резкое увеличение избыточной аппаратуры.

Если требования к системе таковы, что нет необходимости локализовать ошибки с точностью до модуля, и ве-

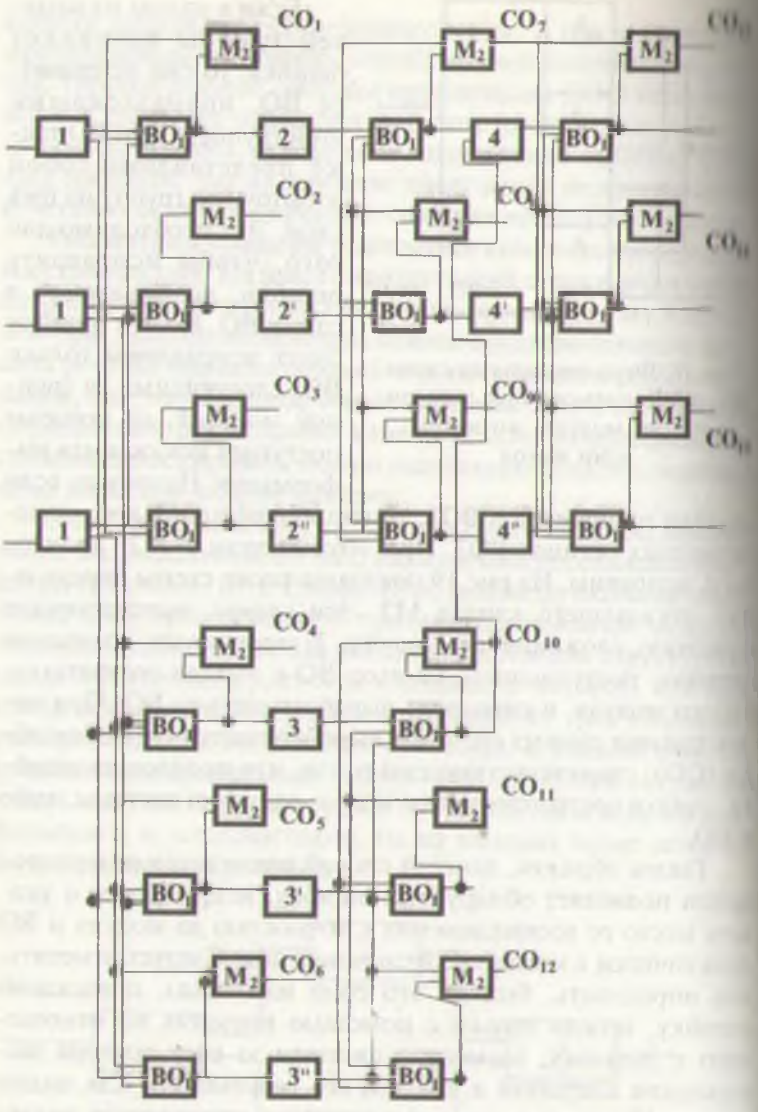


Рис. 19. Структурная схема избыточного элемента системы, в котором каждый исходный модуль представлен тремя одинаковыми, работающими в идентичных режимах

вероятность одновременного появления ошибки более чем в одном модуле избыточного элемента чрезвычайно мала [illegible] одновременность появления ошибки в нескольких определенных модулях может быть и в том случае, если в момент времени t_1 возник отказ одного модуля, в момент t_2 – отказ или сбой другого, причем за интервал $[t_1, t_2]$ модуль не успели восстановить), то можно применить общее для элемента мажорирование, которое показано на рис. 20. Из рассмотрения этой схемы видно, что ошибка, происходящая в одном элементе будет исправлена (возможна также ошибка одновременно в любой совокупности модулей данного элемента), будет обнаружена и локализована с точностью до элемента схемами ВО и М2. Преимущества данного способа реализации мажорирования – резкое сокращение схем ВО и М2 по сравнению с первым вариантом. Недостатком его является то, что при отказе любого модуля необходимо заменить элемент системы в целом, при этом ошибки из-за одновременных сбоев и отказов разных модулей в разных элементах системы (например 2 и 3′, 4 и 1″ и т.д.), как это возможно в первом варианте, не исправляются.

Таким образом, первый вариант лучше применять в системах, в которых возможно накопление отказов. Так, если в момент времени t_1 отказал модуль 1, то избыточный элемент системы будет находиться в работоспособном состоянии и выдавать правильный результат обработки, затем, если в момент времени t_2 отказал модуль 2′, элемент также сохранит работоспособное состояние и т.д. Элемент откажет в том случае, если в отказовом состоянии будут находиться хотя бы два одинаковых модуля (1′ и 1″, 2′ и 2″ и т. д.).

В ОУВС, построенных на базе микропроцессорных наборов, находит распространение такой вид постоянного резервирования, как параллельное. При этом в качестве средства обнаружения ошибок используется дублирование со сравнением результатов обработки. В этом случае уровень детализации, на котором происходит дублирование, целесообразнее выбирать, исходя из конструктивных особенностей вычислительных систем. Например, на одном типовом элементе замены (ТЭЗ) размещать основной элемент, резервный и схему сравнения. Такой способ органи-

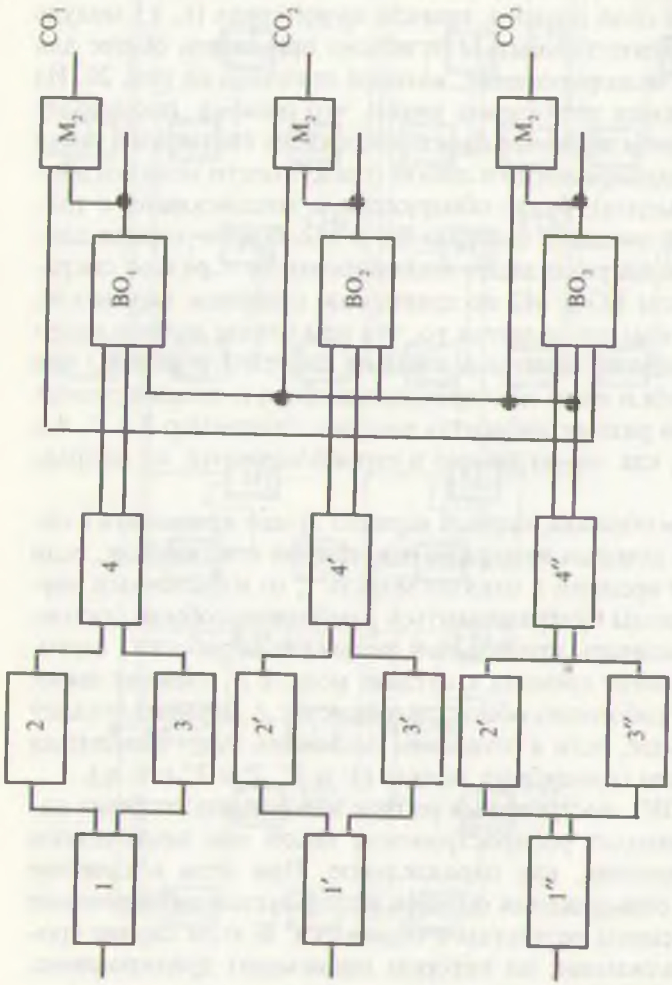


Рис. 20. Схема для общего элемента мажорирования

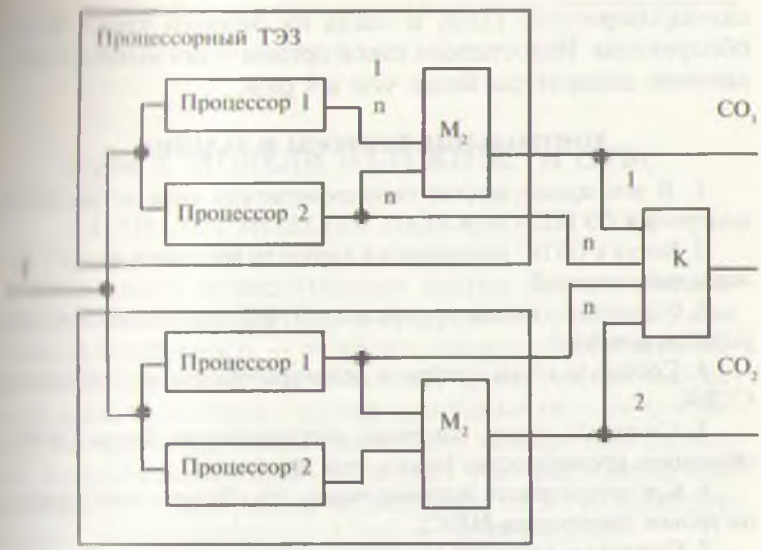


Рис. 21. Схема организации параллельного резервирования на уровне процессора

ции обнаружения ошибок характеризуется простотой организации (практически не требует разработки специальных схем) и по затратам аппаратуры в микропроцессорном исполнении ненамного отличается от схем контроля по модулю. Разработка стандартных ТЭЗ с внутренним дублированием схем позволяет просто решить вопрос об исправлении ошибок. Для этого необходимо дублировать сами ТЭЗы и использовать коммутатор (К), который по сигналам ошибок со схем сравнения (М2) в ТЭЗах выдает информацию с ТЭЗа, не фиксирующего ошибки. Такая организация параллельного резервирования на уровне процессора показана на рис. 21. Она позволяет обнаружить факт возникновения ошибки в ТЭЗе и, следовательно, локализовать ее до сменного элемента в системе и маскировать одиночную ошибку (здесь этот термин понимается как любая ошибка, обнаруженная схемой сравнения в одном ТЭЗе). Следует заметить, что в этом случае просто решается вопрос об отключении отказавшего ТЭЗа и его замене на исправный. Элементы M_2 и К можно выполнить в виде

самопроверяемых схем, и тогда их ошибки также будут обнаружены. Недостатком такой организации является увеличение аппаратуры более чем в 4 раза.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. В чем преимущество самопроверяемых схем встроенного контроля в ОУВС?
2. Когда в ОУВС выполняется алгоритм восстановления — безопасный останов?
3. С помощью какого устройства в ОУВС производится маскирование ошибок?
4. Составьте схему контроля дешифратора с использованием ССВК.
5. Составьте схему контроля операционного блока ОУВС, объясните преимущества такого вида контроля.
6. Как организовать резервирование для обнаружения ошибок на уровне процессора МПС?
7. Приведите алгоритм восстановления вычислительного процесса в ОУВС после возникновения мягкой ошибки.
8. Приведите схему восстанавливающего органа ОУВС.
9. Покажите пример организации параллельного резервирования на уровне процессора в ОУВС.
10. Какие виды ошибок в ОУВС можно исправлять с помощью маскирования?

Глава 4. МОДЕЛИ НАДЕЖНОСТИ ОУВС

4.1. АНАЛИЗ МОДЕЛЕЙ НАДЕЖНОСТИ ОУВС

Надежность вычислительных систем зависит в основном от отказов аппаратуры и программного обеспечения, а отказоустойчивость — от процессов восстановления. Надежность аппаратуры в настоящее время достаточно исследована. Существуют вполне пригодные методы для увеличения прогнозирования надежности электронных элементов аппаратуры, которые позволяют учитывать различные режимы и условия работы элементов, а также их конструктивные, технологические и другие особенности.

Методы оценки программного обеспечения к настоящему времени достигли такого уровня развития, который позволяет прогнозировать надежность программ на основе достаточного количества экспериментов наблюдений.

Заметно отстает решение проблемы оценки процессов восстановления. В первую очередь это обусловлено тем, что автоматическое восстановление является новым подходом в создании ОУВС. Кроме того, процессы восстановления отличаются большим разнообразием, что в свою очередь вызывает затруднения при построении моделей надежности ОУВС.

Первые работы по созданию ОУВС связаны с именем А. Авиженниса. Наиболее известной является его обобщенная модель надежности ОУВС. Эта, а также другие существующие модели надежности ОУВС в основном построены по следующей схеме [10, 14].

Предполагается, что ОУВС может быть представлена как совокупность однородных подсистем. Процессоры, линии связи, блоки памяти, периферийные устройства могут рассматриваться в качестве таких систем.

При этом часть подсистем i-го ($i = 1,n$) типа находится в активном состоянии, а остальные — в резерве. При отказе j-й подсистемы i-го типа, находящейся в активной час-

ти системы, в результате реконфигурации система перехо-
дит в иное состояние с числом резервных подсистем i-го
типа на единицу меньше. Граф состояний такой системы
($n = 2$) изображен на рис. 22, где λ_1 – интенсивность отка-
зов подсистем первого типа, а λ_2 – интенсивность отказов
подсистем второго типа: r, k – значения количества подси-
стем каждого типа соответственно; $l_{a,b}$ – состояние системы,
когда отказали a подсистем первого и b подсистем второго
типа ($0 \le a \le r$, $0 \le b \le k$).

Модель Авижиениса и подобные ей модели учитывают
только изменение количества резервных подсистем, т.е.
восстановление ОУВС при этом происходит только благо-
даря включению резерва.

Так в работе [17] рассматривается отказоустойчивая
многопроцессорная матрица, выполненная в виде сверх-
большой интегральной схемы СБИС. Расчет надежности
производится на основании Марковской модели с учетом
постепенной «деградации» отдельных компонентов матри-
цы (подобно модели Авижиениса).

Таким образом, в существующих моделях надежности
ОУВС не учитываются конкретные способы контроля и вос-
становления. Однако практически в каждой из вычисли-
тельных систем имеются мажоритарные способы резерви-
рования, а также другие аппаратные и программные спосо-
бы контроля и восстановления ОУВС.

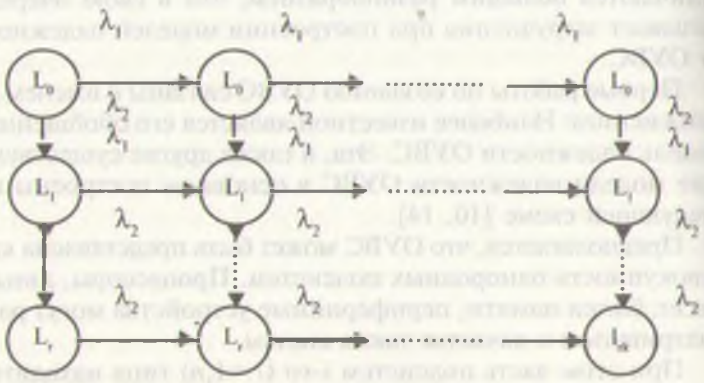


Рис. 22. Граф состояний и переходов ОУВС ($n = 2$)

В работах [20, 21] отражены вопросы построения обобщенной модели надежности ОУВС класса замкнутых систем. Показано, что различие между замкнутыми (закрытыми, неремонтируемыми) и открытыми (обслуживаемыми, ремонтируемыми) ОУВС заключается в том, что в замкнутых системах не предполагается внешнего вмешательства (ремонта). Построение и расчет моделей надежности ремонтируемых ОУВС сложнее, чем неремонтируемых, так как при этом учитываются не только параметры процессов самовосстановления, но и стратегия (дисциплина) ремонта, и в общем случае модели неремонтируемых ОУВС можно считать частными случаями моделей ремонтируемых ОУВС.

В статье [29] вероятность отказа системы рассматривается как сумма условных вероятностей свершения событий, имеющих место в процессе работы ОУВС и ведущих к отказу системы. Однако практическое определение значений условных вероятностей весьма проблематично.

Во многих других работах отражены существующие и наиболее перспективные направления развития ОУВС, способы контроля, диагностирования и восстановления ОУВС.

Необходимо отметить также ряд работ, посвященных построению моделей надежности многопроцессорных ВС, в которых использованы различные архитектурные решения ОУВС [7, 14, 18]. При этом основными критериями меры устойчивости системы считаются: максимальное число неисправных элементов, при котором система может продолжать функционировать, и снижение производительности передачи и обработки информации за счет удлинения путей вследствие отказа отдельных элементов.

Таким образом, существует большое количество разнообразных методов обеспечения отказоустойчивости. Для их рационального сочетания необходимы эффективные модели количественного анализа ОУВС.

4.2. ВЫБОР КРИТЕРИЕВ ОПТИМАЛЬНОСТИ

По существу, проектирование ОУВС — задача многокритериальной оптимизации. Основными критериями оптимальности ОУВС являются производительность системы, ее надежность (безотказность) и сложность системы, вы-

ражаемая через массу, габариты, стоимость, количество элементов, объем ЗУ и др.,

$$R(t) = F(П, T, C, M, ..., P, Q).$$

Обычно один из критериев оптимальности выбирают в качестве основного, а на другие накладывают ограничения. Так, например, возможны требования максимальной производительности при ограничениях на надежность и сложность, максимальной надежности при ограничениях на производительность (снизу) и сложность, минимальной сложности при ограничениях на производительность и надежность.

Интерес представляет также подход, заключающийся в сведении многокритериальной задачи к однокритериальной.

Наиболее общим критерием, характеризующим возможности ВС, является производительность. Идеальная производительность $П$ может быть выражена через количество некоторых условных операций преобразования информации в единицу времени при отсутствии нарушения вычислительного процесса. Поскольку в процессе работы ВС могут возникнуть ошибки, то фактически производительность $П_ф$ меньше идеальной. Допустим, что значения $П_0$ $П_ф$ связаны следующим линейным уравнением:

$$П_ф = g \cdot П_0 + h, \qquad (4.1)$$

где g, h — коэффициенты, зависящие от показателей надежности элементов системы, от степени отказоустойчивости и от других факторов. В простейшем случае $h = 0$, g — относительное число неискаженных из-за нарушения вычислительного процесса результатов операции. Отметим, что здесь в дальнейшем под нарушениями вычислительного процесса подразумевается отказ элемента ВС, сбой, ошибка в программе, ошибка оператора и др.

Пусть критерий оптимальности учитывает также сложность отказоустойчивой системы C. Тогда относительная производительность

$$П = П_ф / C. \qquad (4.2)$$

Следовательно, для нахождения оптимального решения остается определить параметры ОУВС, обеспечивающие максимум $П$.

Пример.

Пусть дана *ВС*, состоящая из n центральных процессоров со сложностью (стоимостью) $C_п$ и производительностью $П_{оп}$ и m периферийных устройств со сложностью (стоимостью) C и производительностью $П_{от}$. Для простоты будем считать, что в идеальной системе $П_0 = \min\{nП_{оп}; mП_{от}\}$, т.е. каждая операция обработки связана с обращением к периферийному устройству, а производительность системы определяется производительностью той группы устройств, пропускная способность которых меньше. В общем случае, когда на r операций обработки приходится одна операция периферийного устройства, производительность последнего увеличивается в r раз.

Пусть известны интенсивность отказов $\lambda_п$ и интенсивность восстановления $\mu_п$ центрального процессора и соответствующие показатели $\lambda_т$ и $\mu_т$ устройства. Тогда, допуская, что фактическая производительность системы определяется числом неискаженных операций, имеем

$$П_ф = \min\left\{n \cdot П_{оп}\frac{\mu_п}{\mu_п + \lambda_п};\ m \cdot П_{от}\frac{\mu_т}{\mu_т + \lambda_т}\right\},$$

а относительная производительность:

$$П = \frac{П_ф}{n \cdot C_п + m \cdot C_т}.$$

Задача заключается в нахождении n и m, обеспечивающих максимум $П$ при заданной $П_ф$. Пусть

$П_{оп} = 8$ ч$^{-1}$	$П_{от} = 4$ ч$^{-1}$	$C_п = 5$ ед.
$C_т = 2$ ед.	$\lambda_п = 0{,}01$ ч$^{-1}$	$\mu_п = 1{,}0$ ч$^{-1}$
$\lambda_т = 0{,}02$ ч$^{-1}$	$\mu_т = 0{,}1$ ч$^{-1}$	$П_ф = 23$ ч$^{-1}$

Два варианта решения задачи, близкие к оптимальному, приведены в табл.5. Сравнивая их, делаем вывод, что лучшим является вариант $(n, m) = (4,8)$.

Для систем с большим числом разнотипных устр[ойств] оптимальное решение необходимо искать с помощью [спе]циальных методов целочисленной оптимизации.

Таблица

$n * m$	$П_ф$	$П$
4,8	32.0,99; 32.0,83= 26,6	0,739
3,9	24.0,99; 36.0,83 — 23,8	0,721

В рассмотренном примере фактическая производительность системы определяется количеством правильно выполненных операций, однако это справедливо лишь при следующих условиях:

— результат каждой операции имеет самостоятельное значение;

— ошибка в результате обнаруживается сразу;

— ошибочный результат не имеет других отрицательных последствий, кроме необходимости повторения операции;

— ущерб зависит не от продолжительности простоя, а от количества потерянных операций.

Эти условия выполняются только при простых и малоответственных вычислениях, которые, кроме того, легко контролируются. В этом случае величина определяется как коэффициент готовности системы:

$$g = \frac{\mu}{\mu + \lambda},$$

где μ — интенсивность восстановления; λ — интенсивность отказов аппаратуры.

Если, кроме отказов аппаратуры, учитывается также и сбой, то g определяется как:

$$g = \frac{\mu}{\mu + \lambda} \cdot \frac{1/\tau_{сб}}{\lambda_{сб} + 1/\tau_{сб}},$$

где $\lambda_{сб}$ — интенсивность сбоев; $\tau_{сб}$ — среднее время восстановления системы после сбоя.

Формула справедлива при допущении, что сбои и отказы независимые случайные события.

Для некоторых задач сбои и отказы влекут за собой большие продолжительные операции, связанные с повторением большой части вычислительного процесса, поэтому в общем случае

$$g = \prod_{i=1}^{n} \frac{1}{1+\lambda_i \cdot \tau_i} \approx 1 - \sum_{i=1}^{n} \lambda_i \cdot \tau_i, \qquad (4.3)$$

где λ_i – интенсивность отказов (сбоев) i-типа; τ_i – среднее время восстановления системы после отказа i-го типа с учетом возможности повторения вычислений.

Часто, особенно в управляющих вычислительных системах, задержка в обработке информации тем опаснее, чем продолжительнее эта задержка.

Тогда

$$g = \prod_{i=1}^{n} \frac{1}{1+\lambda_i \cdot f_i(\tau_i)} \approx 1 - \sum_{i=1}^{n} \lambda_i \cdot f_i(\tau_i), \qquad (4.4)$$

где $f_i(\tau_i)$ – некоторая выпуклая вниз нелинейная функция, характеризующая опасность длительных задержек.

Приближенные выражения (4.3) и (4.4) справедливы для $\lambda_i\tau_i<1$.

Сказанное выше относится к ремонтируемым ОУВС, т.е. к системам, в которых, кроме автоматического восстановления, производится замена (ремонт) отказавших компонентов вручную. Для неремонтируемых ОУВС это справедливо лишь в том случае, если вместо показателей производительности $П$, $П_ф$, $П_∞$ рассматривать соответственно относительную долговечность T, фактическую долговечность $T_ф$ [(как среднее время до окончательного отказа ОУВС с учетом реальных возможностей автоматического восстановления вычислительного процесса и идеальную долговечность (как среднее время до окончательного отказа ОУВС, определяемого только отказами резервов).

4.3. МОДЕЛЬ НАДЕЖНОСТИ ОУВС НА ОСНОВЕ ЦЕПЕЙ МАРКОВА С НЕПРЕРЫВНЫМ ВРЕМЕНЕМ

В 4.1. критерием оптимальности ОУВС была выбрана относительная производительность $П$ или относительная достоверность $Т$. Если времена перехода ВС из одного состояния в другое распределены по показательному закону, то для определения значений $П$ и $Т$ может быть применена модель Маркова.

Построим модель состояний ОУВС на основе анализа возможных состояний системы (см. рис. 22), причин возникновения сбоев и отказов компонентов, способов их обнаружения и локализации, а также способов восстановления системы в целом. Различные состояния ОУВС определяются событиями, имеющими место в процессе ее функционирования.

Результаты статистических исследований указывают на то, что часто распределение времени безотказной работы вычислительных систем близко к экспоненциальному закону. Кроме того, из практических расчетов следует, что в задачах надежности замена неэкспоненциального распределения времени восстановления на экспоненциальное приводит к незначительной погрешности. Поэтому с учетом [14, 18, 21] можно построить модель надежности ОУВС на основе цепей Маркова с непрерывным временем.

Введем следующие обозначения состояний ОУВС:

e_1 – полностью или частично работоспособное состояние;

e_2 – в системе имеется ошибка (сбой или отказ);

e_3 – ошибка обнаружена;

e_4 –ошибка не обнаружена;

e_5 – ошибка локализована;

e_6 – ошибка не локализована;

e_7 – повторение последней операции;

e_8 – ошибка маскирования;

e_9 – возвращение системы к контрольной точке;

e_{10} – информация восстановлена;

e_{11} – повторение (продолжение) вычислительного процесса, начиная с контрольной точки;

e_{12} – отказавший модуль удален;

E_{13} – резервный модуль введен;
E_{14} – резервы системы исчерпаны;
E_{15} – во время восстановления работы системы возникла другая ошибка;
E_{16} – автоматические средства восстановления отказали;
E_{17} – средства коммутации резервов отказали;
E_{18} – ОУВС отказала.

Модель состояний ОУВС можно представить в виде графа (рис. 23). Эта модель удобна тем, что позволяет непосредственно определять значения времени и вероятности пребывания ВС в отдельных состояниях путем решения системы дифференциальных уравнений.

Постановка задачи: Пусть дана ОУВС, в процессе функционирования которой могут иметь место перечисленные выше события. Предположим, что потоки событий пуассоновские с параметром μ. Тогда плотность распределения времени пребывания системы в i-м состоянии описывается в виде $f_i = \mu_i \exp(-\mu_i t)$. В каждый конкретный момент времени t система находится в i-м состоянии с вероятностью $P_i(t)$, т.е. $\sum_{i=1}^{18} P_i(t) = 1$. Обозначим интенсивность перехода из i-го состояния в j-е через $\mu_{ij}(i, j = 1,18)$. Интенсивность перехода определяется как $\mu_y = \frac{1}{t_y}$ – среднее время свершения i-го события при переходе системы из i-го в j-е состояние (например, $t_{ср}$ среднее время наработки системы на отказ или сбой).

На основе сделанных выше допущений найдем вероятности пребывания системы в каждом из возможных состояний $P_i(t)$ $(i = 1,18)$.

Решение задачи. По методу цепей Маркова с непрерывным временем вероятность $P_i(t)$ того, что система находится в i-м состоянии в момент времени t, определяется путем решения системы линейных дифференциальных уравнений, составленных на основании графа состояний и переходов ОУВС (рис. 23):

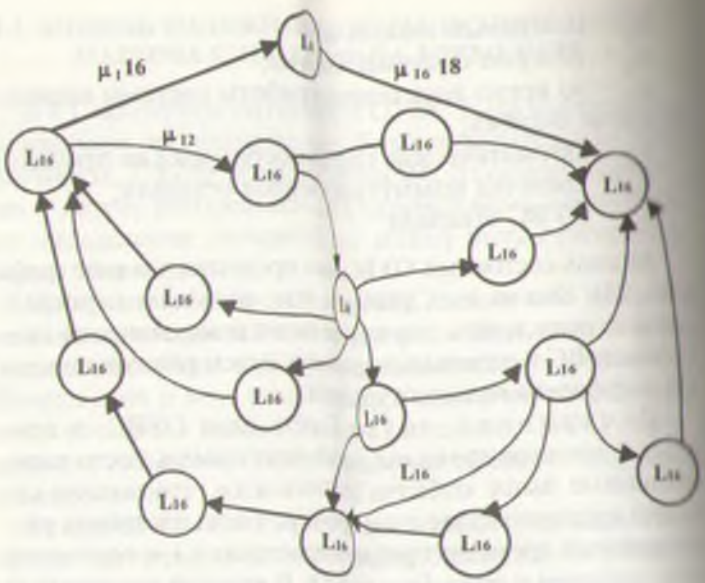


Рис.23. Граф состояний и переходов ОУВС с учетом процесса восстановления

$$\frac{P_1(t)}{dt} = -(\mu_{1,2} + \mu_{1,16})P_1(t) + \mu_{2,1}P_2(t) + \mu_{8,1}P_8(t) + \mu_{11,1}P_{11}(t);$$

$$\frac{P_2(t)}{dt} = -(\mu_{2,3} + \mu_{2,4})P_2(t) + \mu_{1,2}P_1(t);$$

$$\frac{P_3(t)}{dt} = -(\mu_{3,7} + \mu_{3,8} + \mu_{3,5} + \mu_{3,6} + \mu_{3,15})P_3(t) + \mu_{2,3}P_2(t);$$

$$\frac{P_4(t)}{dt} = -\mu_{4,18}P_4(t) + \mu_{2,4}P_2(t);$$

$$\frac{P_5(t)}{dt} = -(\mu_{5,9} + \mu_{5,12})P_5(t) + \mu_{2,5}P_2(t);$$

$$\frac{P_6(t)}{dt} = -\mu_{6,18}P_6(t) + \mu_{3,6}P_3(t);$$

$$\frac{dP_1(t)}{dt} = -\mu_{1,1} P_1(t) + \mu_{3,7} P_3(t); \qquad (4.3.1)$$

$$\frac{dP_3(t)}{dt} = -\mu_{3,3} P_3(t) + \mu_{3,8} P_3(t);$$

$$\frac{dP_5(t)}{dt} = -\mu_{9,10} P_9(t) + \mu_{5,9} P_5(t) + \mu_{12,9} P_{12}(t) + \mu_{13,9} P_{13}(t);$$

$$\frac{dP_{10}(t)}{dt} = -\mu_{10,11} P_{10}(t) + \mu_{9,10} P_9(t);$$

$$\frac{dP_{11}(t)}{dt} = -\mu_{11,1} P_{11}(t) + \mu_{10,11} P_{10}(t);$$

$$\frac{dP_{12}(t)}{dt} = -(\mu_{12,9} + \mu_{12,13} + \mu_{12,14} + \mu_{12,17}) P_{12}(t) + \mu_{5,12} P_5(t);$$

$$\frac{dP_{13}(t)}{dt} = -\mu_{13,9} P_{13}(t) + \mu_{12,13} P_{12}(t);$$

$$\frac{dP_{14}(t)}{dt} = -\mu_{14,18} P_{14}(t) + \mu_{12,14} P_{12}(t);$$

$$\frac{dP_{15}(t)}{dt} = -\mu_{15,18} P_{15}(t) + \mu_{3,15} P_3(t);$$

$$\frac{dP_{16}(t)}{dt} = -\mu_{16,18} P_{16}(t) + \mu_{1,16} P_1(t);$$

$$\frac{dP_{17}(t)}{dt} = -\mu_{17,18} P_7(t) + \mu_{12,17} P_{12}(t);$$

$$\frac{dP_{18}(t)}{dt} = -\mu_{16,18} P_{16}(t) + \mu_{4,18} P_4(t) + \mu_{15,18} P_{15}(t) + \mu_{6,18} P_6(t) +$$

$$\mu_{14,18} P_{14}(t) + \mu_{17,18} P_{17}(t).$$

При условии нормировки $\sum_{i=1}^{18} P_i(t) = 1$. уравнение (4.31) в матричной форме имеет вид

$$\frac{dP(t)}{dt} = P(t) \cdot M,$$

где $P(t) = [P_1(t), P_2(t), \ldots P_{18}(t)]$, а значение интенсивности восстановления μ_{ij} соответствует элементу i-й строки j-го столбца квадратной матрицы М. Диагональные элементы матрицы М находятся из условия, что сумма элементов строки должна равняться нулю.

Переходя к изображению по Лапласу (обозначение звездочкой), и полагая, что $P_1(t = 0) = 1$, $P_i(t = 0) = 0$, $i = 1, 18$, систему уравнений (4.31) можно решить как алгебраическую. Тогда вероятность безотказной работы системы составит

$$P_1(S) = \left[(S + \mu_{1,2} + \mu_{1,16}) - \frac{\mu_1 * \mu_{2,3} * \mu_A}{(S + \mu_{3,7} + \mu_{3,8} + \mu_{3,6} + \mu_{3,5}) * (S + \mu_{2,3} + \mu_{2,4})} \right]^{-1},$$

где S – оператор Лапласа.

$$\mu_A = \frac{\mu_{7,1} * \mu_{3,7}}{S + \mu_{7,1}} + \frac{\mu_{8,1} * \mu_{3,8}}{S + \mu_{8,1}} + \frac{\mu_{11,1} * \mu_{10,11} * \mu_{9,10}}{(S + \mu_{11,1}) * (S + \mu_{10,11}) * (S + \mu_{9,10})} \cdot$$

$$\cdot \frac{\mu_{5,9} * \mu_{3,5}}{S + \mu_{5,9} + \mu_{5,12}} +$$

$$+ \frac{\mu_{3,5} * \mu_{5,12} * (\mu_{13,9} * \mu_{12,13} + \mu_{12,19} * (S + \mu_{13,9}))}{(S + \mu_{13,9}) * (S + \mu_{12,9} + \mu_{12,13} * \mu_{12,14} + \mu_{12,17}) * (S + \mu_{5,9} + \mu_{5,12})}$$

$$T_{cp} = P_1(S)_{s=0} = \left[\mu_{1,2} + \mu_{1,16} - \frac{\mu_{2,5} * \mu_{2,4} * \mu_B}{(\mu_{8,7} + \mu_{3,8} + \mu_{3,5} + \mu_{3,6} + \mu_{3,15}) * (\mu_{2,3} + \mu_{2,4})} \right]^{-1},$$

$$\mu_{3,5} + \mu_{3,8} + \frac{\mu_{5,9} * \mu_{3,5}}{\mu_{5,9} + \mu_{5,12}} + \frac{\mu_{3,5} * \mu_{5,12} * (\mu_{12,9} + \mu_{12,13})}{(\mu_{12,9} + \mu_{12,13} * \mu_{12,14} + \mu_{12,17}) * (\mu_{5,7} + \mu_{5,12})}.$$

4.4. ПОЛУМАРКОВСКАЯ МОДЕЛЬ НАДЕЖНОСТИ ОУВС

Аппарат полумарковских процессов дает возможность учитывать непуассоновские потоки переходов в аналитической форме. Для задач теории надежности это существенно, поскольку время восстановления устройств не всегда распределено по показательному закону.

Анализ надежности с использованием аппарата полумарковских процессов состоит из двух основных этапов. На первом этапе определяются условные вероятности $P_{i,j}(t)$:

$$P_{i,j}(t) = \int_0^{\infty} \prod_{\substack{k=1 \\ k \neq i,j}}^{n} [1 - F_{i,k}(t)] * f_{i,j}(t) dt, \qquad (4.4.1)$$

где $F_{i,k}(t)$ – функция распределения случайного времени перехода из i-го состояния в j-е; $f_{i,j}(t)$ – плотность распределения.

Затем согласно [2, 14] находятся стационарные вероятности (частоты) P_i попадания марковского случайного процесса в i-е состояние:

$$P_i = \sum_{j=1}^{n} P_j * P_{ij} \qquad (4.4.2)$$

при условии, что

$$\sum_{i=1}^{n} P_i = 1. \qquad (4.4.3)$$

На втором этапе анализа определяется среднее время m_i пребывания системы в каждом i-м состоянии, которое выражается как интеграл от вероятности того, что система пребывает в i-м состоянии в течение времени t:

$$m_i = \int_0^{\infty} \prod_{\substack{k=1 \\ k=i}}^{n} [1 - F_{ik}(t)] dt. \qquad (4.4.4)$$

Далее находятся вероятности пребывания системы в i-м состоянии с учетом времени пребывания:

$$q_i = \frac{m_i P_i}{\sum_{j=1}^{n} m_j P_j}, \quad (i, j = \overline{1, n}). \qquad (4.4.1)$$

Покажем на примере, что применение полумарковской модели по сравнению с моделью, где время восстановления распределено по показательному закону, значительно улучшает точность результатов.

Пример. Рассмотрим систему с двумя ненагруженными резервами, однотипными с основной подсистемой. Интенсивность отказов подсистемы $\lambda = 10^{-1}$ ч$^{-1}$, восстановление происходит за фиксированное время $t_в = 1$ ч. Граф состояний и переходов этой системы изображен на рис. 24. Матрица вероятностей переходов для данной системы имеет вид:

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ P_{21} & 0 & P_{23} & 0 \\ 0 & P_{32} & 0 & P_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

где:

$$P_{21} = P_{32} = 1 - \int_0^{t_в} e^{-\lambda \cdot t_в} dt = e^{-\lambda \cdot t_в} = Z.$$

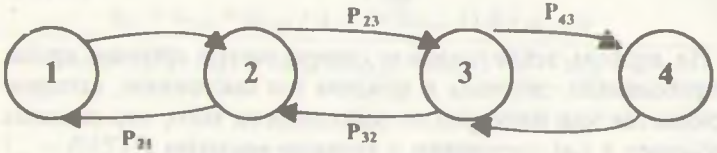


Рис.24. Граф состояний и переходов ОУВС с двумя ненагруженными резервами

Вероятность того, что за время $t_в$ система не перейдет в состояние 3(4), определяется как вероятность противоположного события

$$P_{23} = P_{34} = 1 - e^{-\lambda * t_в} = 1 - Z.$$

Стационарные вероятности находим из уравнений:

$$\begin{aligned}
&-P_1 + P_{21} * P_2 = 0;\\
&P_1 - P_2 + P_{32} * P_3 = 0;\\
&P_{23} * P_2 - P_3 + P_4 = 0;\\
&P_{34} * P_3 - P_4 = 0;\\
&P_1 + P_2 + P_3 + P_4 = 1,
\end{aligned}$$

откуда

$$P_1 = \frac{1}{1 + \frac{1}{Z} + \frac{1-Z}{Z^2} + \frac{(1-Z)^2}{Z^2}} = \frac{1}{2} * \frac{Z^2}{Z^2 - Z + 1};$$

$$P_2 = \frac{1}{2} * \frac{Z^2}{Z^2 - Z + 1};$$

$$P_3 = \frac{1}{2} * \frac{1 - Z}{Z^2 - Z + 1};$$

$$P_4 = \frac{1}{2} * \frac{(1-Z)^2}{Z^2 - Z + 1}.$$

Далее по формуле (4.4.4) получаем:

$$m_1 = \int_0^{\infty} 1 - (1 - e^{-\lambda * t}) dt = \frac{1}{\lambda};$$

$$m_2 = m_3 = \int_0^{t_в} e^{-\lambda * t} dt = (1 - Z) * \frac{1}{\lambda};$$

$$m_4 = t_в.$$

Тогда по формуле (4.4.5) вероятности пребывания системы в состояниях 1, 2, 3, 4 будут:

$$q_1 = Z^2/\eta;$$

$$q_2 = Z*(1-Z)/\eta;$$

$$q_3 = (1-Z)^2/\eta;$$

$$q_4 = (1-Z^2)*t_B*\lambda/\eta;$$

где $\eta = 1 - Z + Z^2 + (1-Z)^2 * t_B * \lambda$.

Когда $\lambda << \mu$ погрешность при использовании марковской модели вместо полумарковской составляет менее 10%. Таким образом, для приведенных выше исходных данных получим следующие результаты. Для полумарковской модели:

$$q_1 = 0{,}8949;$$
$$q_2 = 0{,}09416;$$
$$q_3 = 0{,}009907;$$
$$q_4 = 0{,}0009907.$$

Для модели Маркова:

$$q_1 = 0{,}90009;$$
$$q_2 = 0{,}090009;$$
$$q_3 = 0{,}0090009;$$
$$q_4 = 0{,}00090009.$$

Однако, если, например, $t_B = \frac{1}{\lambda} = 1$, то $Z = e^{-1} = 0{,}368$, $q_1 = 0{,}116$, $q_2 = 0{,}200$, $q_3 = q_4 = 0{,}342$.

Если считать, что время восстановления системы распределено по показательному закону, то $q_1 = q_2 = q_3 = q_4 = 0{,}25$. Отсюда делаем вывод, что более сложную полумарковскую модель целесообразно применять только в случаях, когда требуется высокая точность результата или интенсивность отказов приближается к интенсивности восстановления μ.

Этот вывод подтверждает также метод получения асимптотических оценок А. Д. Соловьева [8, 14] для полумарковских процессов с состоянием регенерации. Состоянием регенерации (началом ремонта) в ОУВС можно считать

ступления первого отказа элемента системы после предыдущего ремонта. Время ремонта, как правило, намного меньше, чем среднее время до наступления следующего отказа элемента. Поэтому вероятность отказа ОУВС $q_{o.p.}$ на одном периоде регенерации – малая величина.

Критерий А.Д. Соловьева имеет вид

$$\alpha_c = \frac{M * \xi^2}{(M * \xi)^2} * q_{o.p.},$$

где $M * \xi^2$ — дисперсия периода регенерации; $M * \xi$ — математическое ожидание периода регенерации (также малая величина). По методу А. Д. Соловьева, вероятность отказа системы определяется по показательной модели с погрешностью не более α_c.

4.5. МОДЕЛИ НАДЕЖНОСТИ РЕМОНТИРУЕМЫХ И НЕРЕМОНТИРУЕМЫХ ОУВС

По модели, приведенной в 4.3, трудно проследить влияние процессов восстановления на надежность системы, поэтому интерес представляют предлагаемые в [4, 5] комбинированные модели. В них раздельно описывается процесс постепенной «деградации» системы за счет отказов ее компонентов и процесс автоматического восстановления системы, характеризуемый вероятностью успешного завершения. Модель неремонтируемых ОУВС основана на временных зависимостях отдельных этапов процесса восстановления. Здесь предполагается, что ОУВС состоит из однотипных подсистем с интенсивностью отказов λ и «скользящего» резерва. При неуспешном автоматическом восстановлении с вероятностью q система переходит в отказовое состояние. Составляется граф состояний и переходов, и на его основании система дифференциальных уравнений.

В случае неремонтируемой ВС особый интерес представляет определение среднего времени безотказной работы. В [14] приведен пример, из которого следует, что по показателю среднего времени безотказной работы неремонтируемая ОУВС малоэффективна даже при полном автоматичес-

ком восстановлении ($q = 0$). При $q \geq 0{,}3$ нецелесообразно увеличение среднего времени безотказной работы системы путем резервирования.

Модель надежности однородной ремонтируемой ОУВС основана на следующих допущениях. Предполагается, что ОУВС состоит их однотипных подсистем со средней интенсивностью отказов λ. Количество состояний и интенсивность отказов в каждом из состояний определяются аналогично тому, как это сделано выше. В случае неуспешного автоматического восстановления с вероятностью q система переходит в отказовое состояние. Кроме того, в отличие от предыдущей модели в системе предполагается ремонт с интенсивностью μ. Допускается, что интенсивность ОУВС не зависит от состояния системы. Такая модель соответствует ОУВС, продолжительность ремонта которой определяется в основном временем доставки резервных подсистем. Далее процедура вычисления параметров безотказности аналогична формуле 4.3.

Если автоматическое восстановление не производится, т.е. $q = 1$, то вероятность пребывания системы в состоянии отказа

$$Q = \frac{\lambda_0}{\lambda_0 + \mu}.$$

Среднее время работы ОУВС между отказами

$$T_0 = \frac{\sum_{i=0}^{N-1} P_i}{q\sum_{i=0}^{N-2} P_i * \lambda_i + P_{N-1} * \lambda_{N-1}} = \frac{1 - Q}{\mu * Q}.$$

Таким образом, чтобы вычислить значения P_i и T_0 по моделям надежности неремонтируемой и ремонтируемой ОУВС, необходимо найти вероятность отказа средств автоматического восстановления q.

Установившееся значение параметра потока отказов системы

$$\omega = \frac{1}{T_0} = \frac{\mu * Q}{1 - Q}.$$

Установившееся значение коэффициента готовности системы

$$K_г = 1 - Q.$$

Из приведенных в [14] примеров следует, что неремонтируемая ОУВС малоэффективна, так как ее показатели надежности немного лучше, чем у нерезервированной системы. В то же время ремонтируемая ОУВС может быть практически безотказна даже в случае невысокой интенсивности ремонта. Но при этом вероятность неуспешного автоматического восстановления q должна быть малой величиной ($q < p$ 0,1). Значение q определяется характеристиками процесса автоматического восстановления, (см. гл. 6).

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Определите основные требования, предъявляемые к моделям надежности ОУВС.
2. Какие задачи решаются при оценке надежности ОУВС?
3. Чем отличаются модели надежности ОУВС от моделей надежности ВС?
4. Составьте обобщенную модель надежности ОУВС (модель Авиженниса) в виде графа состояний и переходов .
5. Что принимается за критерий меры устойчивости в моделях надежности ОУВС?
6. Каким образом можно оценить относительную производительность ОУВС?
7. Составьте систему дифференциальных уравнений для расчета надежности ОУВС, описанной в п. 4.3.
8. Найдите значение вероятности средств автоматического восстановления q для моделей надежности ремонтируемых ОУВС.

Глава 5. МОДЕЛИ ПРОЦЕССОВ КОНТРОЛЯ И ДИАГНОСТИРОВАНИЯ ОУВС

5.1. МОДЕЛИ ПРОЦЕССОВ КОНТРОЛЯ

Средства контроля ОУВС подразделяются на аппаратные, программные и смешанные. Они характеризуются тремя основными параметрами: полнотой контроля, временем обнаружения ошибки и сложностью.

Полнота контроля ОУВС оценивается как доля отказов, обнаруживаемых в результате контроля от общего их количества.

$$\alpha = \frac{\sum_{i \in M_к} \lambda_i * n_i}{\sum_{i \in M} \lambda_i * n_i},$$

где M – множество элементов, подлежащих контролю, $M_к$ – множество всех элементов системы; n_i – число элементов i-го типа; λ_i – интенсивность отказов элементов i-го типа.

Иногда полнота контроля выражается как отношение количества контролируемой аппаратуры к ее общему количеству в системе:

$$\alpha_1 = \frac{\sum_{i \in M_к} n_i}{\sum_{i \in M} n_i}.$$

Время обнаружения ошибки (время контроля) определяется как интервал времени от момента возникновения ошибки до момента ее обнаружения при условии, что ошибка будет обнаружена.

Сложность аппаратных средств контроля характеризуется их массой, размерами, стоимостью, потребляемым током и мощностью. Сложность программных средств контроля определяется необходимыми для выполнения программы аппаратными средствами, памятью для размещения программы и операционными устройствами для ее выполнения. При использовании смешанных средств контроля сложность рассчитывается по следующей формуле:

$$C = C_a + C_{al} * a + \sum_i C_{a2i} * b_i,$$

где C_a – сложность аппаратных средств контроля; C_{al} – сложность операционных устройств, реализующих программу контроля; a – коэффициент, определяющий долю времени, затраченного на реализацию программы контроля; C_{a2i} – сложность i-го запоминающего устройства, применяемого для размещения программных средств контроля; b_i – коэффициент, определяющий долю объема i-го запоминающего устройства программными средствами контроля.

Рассмотрим основные методы контроля, которые реализуются с помощью перечисленных средств.

5.1.1. Аппаратные методы контроля

Контроль дублированием (рис. 25). Суть метода состоит в том, что два идентичных операционных устройства O_1 и O_2 работают синхронно при одинаковых исходных данных. В случае возникновения ошибки в одном из них результаты на выходах O_1 и O_2 будут различаться. Это фиксируется устройством сравнения (М2), которое выдает сигнал об ошибке.

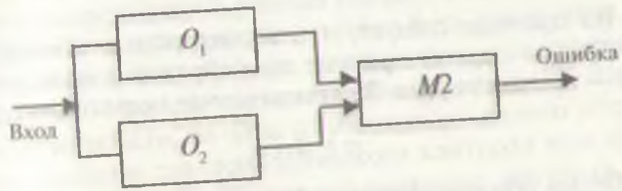


Рис. 25. Схема контроля дублированием

Полнота контроля дублированием приближается к единице. Не обнаружение ошибки может произойти по двум причинам:

1) если в устройствах O_1 и O_2 одновременно возникн одинаковые ошибки;

2) если откажет устройство сравнения.

Условная вероятность того, что возникшая ошибка будет обнаружена, определяется так:

$$q_k = 1 - \left[1 - \frac{\tau * \sum_{i=1}^{n} \lambda_i^2}{\sum_{i=1}^{n} \lambda_i}\right] * \left(1 - \frac{1}{2} * \tau_c * \lambda_c * \nu\right) \approx \frac{\tau * \sum_{i=1}^{n} \lambda_i^2}{\tau * \sum_{i=1}^{n} \lambda_i} + \frac{1}{2} * \tau_c * \lambda_c *$$

где τ — продолжительность такта работы контролируемого устройства; n — число элементов в контролируемом устройстве; λ_i — интенсивность отказов i-го элемента контролируемого устройства; τ_c — время между двумя проверками устройства сравнения; λ_c интенсивность отказов устройства сравнения, ν — условная вероятность того, что в результате отказа устройства сравнения сигнал об его отказе не поступает при условии, что контролируемое устройство отказало. Предполагается, что $\tau_c * \lambda_c << 1$ и $\tau * \lambda_i << 1$.

Пример.

Пусть $\tau = 10^{-6}$с, $\lambda_i = 10^{-6}$ч$^{-1}$ ($i = 1,n$), $\tau_c = 10$ ч, $\lambda_c = 10^{-5}$ч$^{-1}$, $n = 10^3$

Тогда условная вероятность необнаружения ошибки

$$q_k = 10^{-6} * \frac{1}{3600} * \frac{10^{-3} * 10^{-12}}{10^3 * 10^{-6}} + \frac{1}{2} * 10 * 10^{-5} \approx 0{,}5 * 10^{-4}.$$

Из примера следует, что вероятность необнаружения ошибок по первой причине пренебрежимо мала, по второй — незначительна. В данном случае полнота контроля

$$\alpha = 1 - q_k.$$

Время обнаружения ошибки при контроле дублированием определяется тремя составляющими: временем от возникновения отказа до обращения к отказавшему элементу контролируемого устройства, временем от обращения к отказавшему элементу до появления ошибочного результата на выходе устройства и временем срабатывания

ройства сравнения. Пренебрегая первым слагаемым, среднее время обнаружения ошибки в устройстве, состоящем из n элементов, можно вычислить по следующей формуле:

$$\tau_{об} = \frac{\sum_{i=1}^{n} \lambda_i * \tau_i * \eta_i}{\sum_{i=1}^{n} \lambda_i * \eta_i} + \tau_{ср},$$

где τ_i – суммарное время прохождения сигнала от входа i-го элемента контролируемого устройства до его выхода; η_i – относительная частота участия i-го элемента контролируемого устройства в работе этого устройства; $\tau_{ср}$ – время срабатывания устройства сравнения.

Величина η_i определяется с учетом того, что в зависимости от комбинации входных сигналов контролируемого устройства некоторые его элементы могут не участвовать в данной операции, и поэтому ошибки в их работе не проявляются на выходе. Для простейшего устройства значения τ_i, η_i могут быть найдены путем анализа его отдельных состояний, для более сложного – с помощью имитационного или натурного моделирования [14, 23]. Приведенные выше соотношения справедливы как для перемежающихся или постоянных отказов, так и для сбоев.

Недостатком метода контроля дублированием является необходимость использования большего количества дополнительной аппаратуры. Оно может быть уменьшено, если осуществлять контроль не непрерывно, а периодически. Например, если система состоит из однотипных операционных устройств, то одно из них или несколько устройств могут служить для периодического контроля всех остальных. Получается нечто типа «скользящего» контроля. Однако при этом требуется дополнительная коммутационная аппаратура. Структурная схема «скользящего» контроля дублированием показана на рис. 26, где 01...ON – операционные устройства; K_1 ... K_m – контролирующие устройства, подключаемые к операционным с помощью коммутатора K.

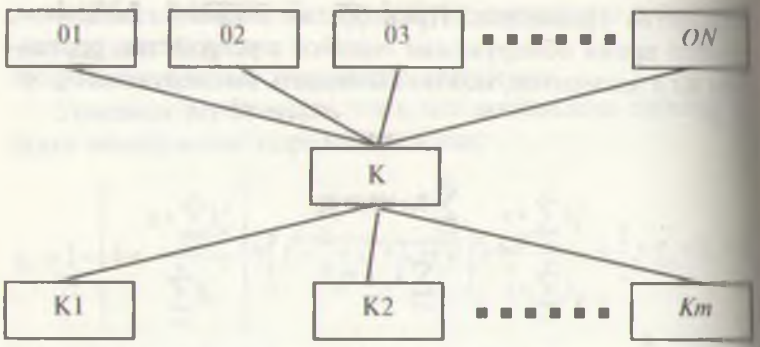


Рис. 26. Схема «скользящего» контроля дублированием

Полнота контроля при «скользящем» контроле дублированием такая же, как и при контроле с полным дублированием, однако среднее время обнаружения ошибки увеличивается за счет времени ожидания очередного подключения контролирующего устройства.

Основным недостатком «скользящего» контроля является то, что полный контроль обеспечивается только при постоянных отказах контролируемого устройства. В случае перемежающихся отказов и особенно при сбоях полнота контроля резко уменьшается и приближается к $\alpha = m/n$, так как сбои, возникающие в периоды, когда данное устройство не контролируется, остаются необнаруженными.

Кроме рассмотренных основных характеристик средств контроля, интерес представляет характеристика, показывающая возможность «ложной тревоги», возможность срабатывания устройства контроля при отсутствии ошибок. Интенсивность ложных срабатываний устройства контроля определяют по формуле

$$\lambda_л = \lambda_{ок} + (1 - \nu) * \lambda_к,$$

где $\lambda_{ок}$ – суммарная интенсивность отказов дублирующих устройств, применяемых для контроля; $\lambda_к$ – суммарная интенсивность отказов устройств сравнения; $(1 - \nu)$ – вероятность того, что в результате отказа устройства сравнения на его выходе появится сигнал об ошибке (ложная единица).

При контроле с помощью специальных кодов важное место в ОУВС занимают коды с обнаружением и исправлением ошибок. Они предназначены для придания устойчивости к отказам и сбоям, устройствам хранения и передачи информации, т.е. устройствам, где информация не преобразуется. Попытки использовать эти коды для обеспечения отказоустойчивости всей ВС пока не увенчались успехом.

5.1.2. Программно-логические методы контроля

Метод контрольных функций. Данный метод заключается в выборе каких–либо дополнительных функций, вычисляемых наряду с основными функциями, и позволяет проверить правильность вычисления последних по определенным соотношениям, называемым контрольными соотношениями.

Например, при решении системы дифференциальных уравнений, суммируя их, можно составить дополнительное дифференциальное уравнение, которое в случае отсутствия ошибок должно также удовлетворяться. Если по реализуемому алгоритму необходимо вычислить k независимых функций, то в целях контроля можно составить дополнительные уравнения в виде линейных комбинаций исходных функций, которые должны удовлетворять определенным контрольным соотношениям. Характеристики программно-логического контроля зависят от конкретных алгоритмов и определяются в каждом случае отдельно.

Полнота программно-логического контроля в общем случае приближается к единице, так как практически любая ошибка приводит к нарушению контрольных функций или соотношений типа равенства. Только когда контролируемый вычислительный процесс содержит неравенство или когда контроль осуществляться по принадлежности результата заданной области, контроль будет неполным. Контроль может осуществляться по соответствию фактического времени выполнения вычислительного процесса заранее рассчитанному (как правило, ошибки в вычислениях приводят к изменению длительности вычислительного процесса, обусловленному остановкой, зацикливанием и т.д.), при этом он будет неполным.

Таким образом, в общем случае полнота контроля

$$\alpha = \text{вер}(x^* \in M_x / x \in M_x),$$

где x^* – значение контролируемого параметра при наличии ошибки; M_x – множество допустимых значений контролируемого векторного или скалярного параметра; x – значение контролируемого параметра при отсутствии ошибки.

Время обнаружения ошибки $\tau_{об}$ при использовании программно-логического контроля складывается из времени появления ошибочного результата на выходе в контролируемой точке модуля или программы $\tau_п$, времени вычисления контрольной функции $\tau_{к.ф}$ и времени сравнения результатов $\tau_{ср}$.

Отдельные слагаемые в этой зависимости определяются временем прохождения соответствующих программ.

Сложность программно-логического контроля определяется объемом памяти, занимаемым программами, которые предназначены для реализации контрольных функций и соотношений, а также долей участия операционных устройств в осуществлении контроля. Доля определяется количеством аппаратуры и частотой ее использования. В общем случае

$$C = \sum_i C_i * \nu_i,$$

где C_i — сложность i-й части аппаратуры; ν_i – частота использования аппаратуры для выполнения функций контроля.

Тестовый контроль. Этот метод широко практикуется для контроля ВС. Полнота тестового контроля может быть приближена к единице, причем число отдельных тестов не превышает количество возможных одиночных отказов типа постоянного нуля и постоянной единицы в контролируемом устройстве, а в большинстве случаев значительно меньше. Время тестового контроля определяется количеством и продолжительностью отдельных тестов. Однако тестовый контроль имеет и недостатки: большое время обнаружения отказа, обусловленное периодичностью контроля; снижение производительности системы вследствие прекращения

ры во время тестирования; необнаружение сбоев, возникающих во время работы системы, когда тестирование не проводится. Из-за названных причин тестовый контроль представляет меньший интерес применительно к ОУВС.[8].

5.2. МОДЕЛИ ПРОЦЕССОВ ТЕХНИЧЕСКОГО ДИАГНОСТИРОВАНИЯ

Цель технического диагностирования аппаратуры ЭВС установить (локализовать) местонахождение неисправного блока модуля, типового элемента замены (ТЭЗ) или другой подсистемы, которая в процессе автоматической реконфигурации исключается из работы системы и затем заменяется в ходе ремонта.

Средства диагностирования также, как и средства контроля могут быть аппаратные, программные и смешанные. При диагностировании, как правило, применяются программные средства, поскольку процедуры диагностирования достаточно сложные. Диагностирование основывается в первую очередь на диагностических тестах, проводимых после обнаружения неисправности. В известной мере в процессе диагностирования могут участвовать средства контроля.

Основными характеристиками средств диагностирования являются глубина диагностирования, время диагностирования и сложность.

Глубина диагностирования определяется размером (количеством элементов) подсистемы, отказ которой устанавливается в результате диагностирования. Чем меньше эта подсистема, т.е. чем более точно указывается группа элементов, среди которых находится отказавший элемент, тем больше глубина диагностирования. Выбор глубины диагностирования является компромиссным решением. Увеличение глубины диагностирования, с одной стороны, приводит к увеличению сложности средств и (или) времени диагностирования, с другой стороны, позволяет более эффективно осуществлять автоматическое восстановление системы, т.е. при малом количестве отключаемой аппаратуры малая глубина диагностирования влечет за собой необходимость отключения в ходе реконфигурации ВС боль-

шого количества основных операционных устройств значительно снижает производительность системы [27].

В аналитической форме средняя производительность системы записывается как

$$П = \sum_k П_k(C_0, C_k, C_Д, C_Р) * P_k(t),$$

где $П_к$ – производительность системы в k-м состоянии; $P_k(t)$ – вероятность нахождения системы в k-м состоянии в момент времени t. Суммарная сложность средств восстановления

$$C = C_0 + C_k + C_Д + C_р,$$

где C_0 – сложность основных средств; C_k – сложность средств контроля; $C_Д$ – сложность средств диагностирования; $C_р$ – сложность средств реконфигурации.

Существуют три основных метода диагностирования: поочередный контроль, метод пересечения подмножеств, исключение подмножеств.

Поочередный контроль. Метод заключается в разделении множества элементов системы на непересекающиеся подмножества и в их поочередной или одновременной проверке до нахождения подсистемы, содержащей отказавший элемент. Метод связан с большими затратами времени и средств диагностирования.

Метод пересечения подмножеств. Суть его состоит в том, что подмножество элементов $M_{отк}$, включающее отказавший элемент, определяется как пересечение подмножеств M_i, среди которых может находиться отказавший элемент в результате каждой i-й проверки:

$$M_{отк} = \bigcap_i M_i.$$

Исключение подмножеств. При использовании этого метода подмножество $M_{отк}$ определяется по формуле

$$M_{отк} = M \setminus \bigcup_j M_j,$$

где M – множество всех элементов диагностируемой системы; M_j – подмножество элементов, заведомо исправных по результатам j-й операции диагностирования.

В однородной системе необходимое число операций $N_п$ диагностирования при поочередном контроле определяется по формуле

$$N_п = \left[\frac{\|M\|}{\|M_j\|}\right],$$

где $\|M\|$ – мощность множества M.

При использовании метода пересечения или метода включения подмножеств каждая операция проверки уменьшает мощность подмножества возможного местонахождения ошибки в лучшем случае в два раза, и поэтому количество необходимых операций определяют следующим образом:

$$N_п = \log_2\left[\frac{\|M\|}{\|M_j\|}\right].$$

Для реальных, неоднородных систем, где $\|M_i\| \neq \|M_j\|$, при $i \neq j$ последняя оценка является приближенной, и фактическое количество необходимых проверок увеличивается.

Вопросы аналитического и имитационного моделирования основных характеристик средств диагностирования рассмотрены в [25, 27].

5.2.1. Глубина диагностирования

Глубина диагностирования определяется подмножеством M элементов системы, проверяемых каждым тестом. Чаще всего тестируемый объект представляют в виде направленного графа с двумя полюсами – входным и выходным. Вершины графа изображают отдельные элементы системы, дуги – связи между ними. Каждому тесту соответствует определенный путь (пути) передачи сигналов от входной вершины к выходной. Если отказал элемент, входящий в путь, то это обнаруживается по изменению выходного сигнала. Следовательно, отказавший элемент при-

надлежит данному пути (подмножеству *M*). Для ОУВС, где отказавшие подсистемы сразу изолируются и в дальнейшей работе системы не участвуют, а неисправности обнаруживаются в результате контроля, можно допустить, что задачей диагностирования является выявление отказавшей подсистемы. Здесь и далее, под изолируемой подсистемой (ИП) понимается подсистема, которая устраняется от дальнейшей работы в ходе реконфигурации, если в ней отказал элемент.

Допустим, что граф, изображающий диагностируемую систему, разделен на отдельные подграфы, соответствующие ИП. Тогда совокупность или последовательность диагностических тестов должна быть достаточной для того, чтобы обнаружить отказы в различных ИП. Определение достаточной совокупности тестов можно упростить, если считать, что для установления неисправности в каждой ИП достаточно проведения одного теста. Однако ИП может оказаться сложным объектом (процессором или даже ВС), и тогда для установления того, что именно данная ИП содержит неисправность, потребуется несколько элементарных тестов.

Допустим, что разбивка системы на изолируемые подсистемы и количество тестов, необходимых для проверки каждого ИП, заданы. Тогда общее количество необходимых тестов определяется исходя из возможности их совмещения, т.е. возможности проверки нескольких ИП одной совокупностью тестов. В идеальном случае (однотипные ИП) достаточно одного теста, причем отказавший ИП устанавливается путем анализа результатов теста.

На практике максимальное число тестов обычно равно суммарному количеству тестов, необходимых для проверки каждого ИП в отдельности. Следовательно, целесообразно применять систему, состоящую из однотипных подсистем, тогда количество хранимых в памяти тестов, а также средств для анализа результатов (реакции на тесты) резко уменьшается. Так как современные ВС включают большое количество процессоров и их число постоянно растет, то, по-видимому, перспективно рассматривать ИП в качестве отдельного процессора. При этом желательно производить взаимную проверку процессоров. Существуют различные

суммы таких проверок. Система, позволяющая надежно диагностировать до t неисправных процессоров, называется t-диагностируемой. За исключением особых случаев, когда возможен одновременный отказ нескольких процессоров под воздействием одного общего фактора, с большей вероятностью можно утверждать, что в данный момент времени ОУВС обнаруживает отказ только одного процессора, поскольку ранее отказавшие процессоры уже изолированы или отключены. Поэтому в ОУВС t = 1.

5.2.2. Продолжительность диагностического тестирования

В случае условного тестирования каждый диагностический тест определяется временем его проведения τ_j. Пусть для локализации каждой k-й неисправности необходимо провести тесты, индексы которых образуют множество M_k. Тогда среднее время диагностирования тестом (диагностического тестирования) определяется по формуле

$$T_y = \sum_{k=1}^{N} P_k * \sum_{j \in M_k} \tau_j,$$

где N – количество возможных неисправностей; P_k – вероятность того, что обнаружена k-я неисправность.

Если диагностирование проводится по методу безусловного тестирования, тесты проходят по заранее заданной последовательности. В этом случае среднее время тестирования находится по следующей формуле:

$$T_б = \sum_{j \in M} \tau_j,$$

где M – множество индексов тестов, проводимых в любом случае для локализации неисправности.

Последняя формула справедлива, когда тесты построены по дихотомическому принципу и неисправность локализуется по комбинации результатов всех тестов. Если же диагностирование сводится к поочередной проверке, то время тестирования оценивается по неравенству

$$T_б \le \sum_{j \in M} \tau_i .$$

Однако при этом требуется большое количество тестов, например, если в системе имеется 1024 элемента, то при поэлементном тестировании понадобится не более 1024 тестов, а в случае дихотомического тестирования – всего $N = \log_2 1024$ тестов $= 10$ тестов.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Определите параметры, характеризующие модели процессов контроля и диагностирования ОУВС.
2. В чем суть контроля дублированием в ОУВС?
3. Как определяется время обнаружения ошибки при контроле дублированием?
4. Составьте структурную схему «скользящего» контроля дублированием.
5. В чем заключается метод контрольных функций?
6. Почему тестовый контроль широко распространен в вычислительных системах?
7. Составьте выражение для определения сложности средств восстановления.
8. Чем отличается метод поочередного контроля от других методов контроля в ОУВС?
9. Как выбирается необходимая глубина диагностирования ОУВС?

Глава 6. ПРОЦЕССЫ РЕКОНФИГУРАЦИИ И ВОССТАНОВЛЕНИЯ

6.1. ХАРАКТЕРИСТИКИ ПРОЦЕССОВ

Процессы реконфигурации характеризуются следующими параметрами: сложностью средств реконфигурации, временем реконфигурации и степенью снижения производительности после реконфигурации.

Сложность средств реконфигурации складывается из сложности средств переключения $C_{с.п}$, специально предназначенных для осуществления реконфигурации, и сложности средств системы $C_{с.с.}$ с учетом доли времени $K_р$ потраченного на процесс реконфигурации.

В аналитическом виде сложность средств реконфигурации

$$C_{с.р.} = C_{с.п.} + C_{с.с.} * K_{р.}$$

Время реконфигурации $T_р$ обычно незначительное. Оно определяется временем анализа сложившейся ситуации в системе $T_{а.с.}$, временем принятия решения о способе реконфигурации $T_{п.р.}$ и временем собственно переключения $T_п$

$$T_р = T_{а.с.} + T_{п.р.} + T_{п.}.$$

Степень снижения производительности после реконфигурации v может быть выражена как отношение производительности системы после реконфигурации к исходной производительности, т.е. производительности системы при отсутствии неисправности.

Аналогичными параметрами характеризуется процесс восстановления информации после реконфигурации. Для его осуществления необходимо найти дублирующую запись потерянной информации, возвратиться к соответствующей контрольной точке и продолжить вычисления. Сложность

средств восстановления информации $C_{с.и.}$ определяется с учетом $K_д$ доли объема ОЗУ, необходимого для дублирования информации, $K_и$ – доли времени, потраченного на восстановление информации. В аналитическом виде

$$C_{с.к.} = C_{з.у.} * K_д + C_{с.с.} * K_и.$$

Время восстановления информации $T_{в.и.}$ складывается из времени поиска дублирующей информации, времени передачи этой информации в соответствующий процессор $T_{п.и.}$ и времени повторения вычислений $T_{п.в.}$ от ближайшей неразрушенной контрольной точки до операции, которая была искажена при возникновении ошибки:

$$T_{в.и.} = T_{п.и.} + T_{п.п.} + T_{п.в.}.$$

6.2. АНАЛИЗ СИТУАЦИИ В СИСТЕМЕ И ПРИНЯТИЕ РЕШЕНИЯ

Анализ ситуации в системе и принятие решения заключаются в установлении отказавшей изолированной подсистемы, сравнении вариантов организации работы системы при отключении отказавшей ИП и в выборе наиболее благоприятного варианта. В общем случае работа ОУВС осуществляется по такой схеме: один из процессоров (или их группа) системы получает задание от внешнего источника, частично обрабатывает его и передает для дальнейшей обработки следующему процессору (или процессорам), который поступает аналогичным образом, и так до последнего процессора (или процессоров). При такой организации работы каждый процессор должен обладать сведениями о состоянии системы, т.е. иметь таблицу ресурсов, которая уточняется либо периодически (с достаточно малым периодом), либо по мере возникновения изменений в состоянии системы.

Для этого удобно применять метод сообщений типа «я жив», т.е. сообщений, задаваемых заведомо исправным процессором и прекращающихся или искажающихся в случае

неисправности, задачи между процессорами могут распределяться либо по некоторому оптимальному плану, который составляется входными процессорами по мере появления задач, либо по более простому алгоритму, например, очередная задача передается очередному процессору при условии, что последний исправен и свободен. Выбор одного из этих способов осуществляется по критерию производительности путем сравнения времени, затраченного на решение задачи по простому алгоритму.

При работе по такой схеме проблема реконфигурации возникает в ходе выполнения общей задачи многопроцессорной параллельной обработки информации, причем, с точки зрения распределения задач, отказавший процессор практически приравнивается к занятому. Отметим, что в рамках описанной организации работы системы удобно осуществлять также контроль по времени выполнения задач. Наряду с решением задачи и подпрограммой для нее предыдущий процессор определяет также необходимое время решения. Сравнение фактического времени решения с расчетным и обеспечивает возможность контроля.

6.3. ПРОИЗВОДИТЕЛЬНОСТЬ СИСТЕМЫ С УЧЕТОМ ОТКАЗОВ

Производительность системы с учетом отказов наиболее просто можно оценить как суммарную производительность работоспособных процессоров. Тогда степень снижения производительности определяется как отношение суммарной производительности исправных процессоров к суммарной производительности всех процессоров системы. Однако такая оценка неточна, поскольку производительность системы, как правило, меньше, чем суммарная производительность входящих процессоров.

С другой стороны, отказы отдельных процессоров или других частей системы приводит к нарушению связей между процессорами. Поэтому может оказаться невозможным использование каких-либо частей системы вследствие ограниченного доступа к ним или его отсутствия.

Для моделирования этой ситуации структур
изобразим в виде графа, вершинами которого
функциональные устройства системы (процессоры
минающие устройства, устройства ввода-вывода и
дугами — связи между ними. Если производитель
дельных устройств представить как пропускную
ность, то производительность сети можно оценить
пропускную способность от входных устройств к
ным по известным алгоритмам [4]. Однако при таком
ходе не учитывается возможность разбивки задач на
дачи, обрабатываемые отдельными процессорами,
образование очередей, объем буферных ЗУ. Более
оценку производительности системы $П_i$ в каждом i-м
стоянии можно получить с помощью методов теории
сового обслуживания или путем имитационного мод
рования. Применение этих методов в данном случае
оценке производительности исправных систем аналогич
[4, 14]. Производительность системы определяется как

$$П = \sum_i П_i * P_i,$$

где P_i — вероятность возникновения i-го состояния.

Пример. В качестве простого примера рассмотрим
оценку производительности ОУВС с сетевой организаци
ей, изображенной на рис. 27. Допустим, что ЭВМ, соотве

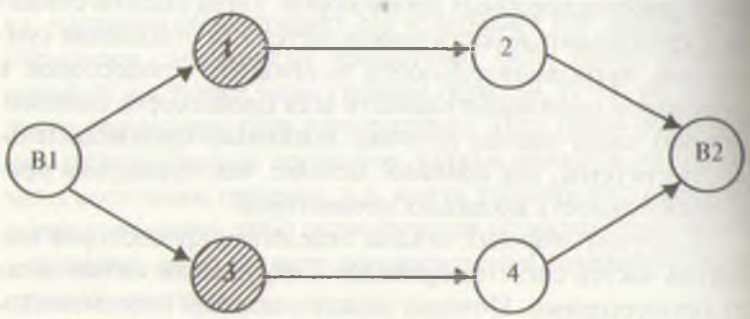


Рис. 27. ОУВС с сетевой организацией работы

...ным вершинам графа (заштрихованные вер-... ...ают устройства ввода B_1, а ЭВМ — соответству-... ...ным вершинам графа, устройства вывода B_2. Дуги ... ...ображают связи между ЭВМ, пригодные для ... данных. В табл. 6 приведены возможные состоя-... ...системы, вероятности их возникновения и соответ-... ...значения производительности.

...полагается, что производительность каждой ЭВМ ... ЭВМ 1 и 2, а также 3 и 4 работают в конвейер-... ...ме с производительностью равной двум. Конвей-... 1, 2, 3 и 4 работают параллельно также с удвоенной ...производительностью.

...ероятность безотказной работы каждого из элементов ...системы $P = 0{,}9$, включая ЭВМ 1, 2, 3 и 4, а также связи ...ередачи данных 1-2, 3-4 между ними, кроме устройств ...вода-вывода, которые считаются безотказными.

Таблица 6

Возможные состояния системы, вероятности их возникновения и соответствующие значения производительности

Состояние системы	Вероятность состояния, P	Производительность, П
Безотказная работа	$0{,}9^6 = 0{,}5314$	4
Отказ верхней подсистемы	$(1 - 0{,}9^3) * 0{,}9^3 = 0{,}1976$	2
Отказ нижней подсистемы	$(1 - 0{,}9^3) * 0{,}9^3 = 0{,}1976$	2
Отказ системы	$(1\text{-}0{,}9^6) = 0{,}07344$	0

Усредненная производительность системы по формуле (6.3.1) имеет вид

$$П = 4 * 0{,}5314 + 2 * 2 * 0{,}1976 = 2{,}916.$$

6.4. МОДЕЛЬ ПРОЦЕССА АВТОМАТИЧЕСКОГО ВОССТАНОВЛЕНИЯ ОУВС

Автоматическое восстановление в вычислительных системах является новым подходом, обеспечивающим высокую

степень надежности, готовности и отказоустойчивости ОУ
Изучению этого вопроса посвящено много работ [14,
28, 31], в каждой из которых трудности построения мо
связаны с большим разнообразием процессов восстано
ния в ОУВС. Ниже предлагается модель, описывающая
щие свойства различных процессов восстановления.

Рассмотрим процесс восстановления некоторой ги
тетической ОУВС, включающей все возможные процесс
восстановления, применяемые в реальных ОУВС [8], а так
множество возможных событий в системе через $S=\{S_1, S_2$
$\ldots S_{14}\}$. Возникшая ошибка в ОУВС (S_1) либо может бы
обнаружена аппаратными средствами контроля (S_2) ил
программными средствами контроля (S_3), или может ос
таться не обнаруженной, в результате чего произойдет от
каз системы (S_4).

Ошибка может быть также маскирована (S_5), если в
системе применяются средства пассивной отказоустойчи
вости. В таком случае вычислительный процесс продолжа
ется без задержки (S_6). В большинстве систем обнаружение
ошибки аппаратными средствами контроля производится
повторным выполнением операции ограниченным числом
раз. Если повторение было успешным (S_7), т.е. имел место
сбой, последствия которого при повторении операции ис
чезли, вычислительный процесс продолжается.

Для возможности повторения операции аппаратные
средства должны сохранить операнды до окончания конт
роля результатов операции. Если повторение операции было
безуспешным (S_8), то это говорит о наличии устойчивого
отказа аппаратуры. В таком случае производится автомати
ческая реконфигурация (S_9), которая заключается либо в
автоматической замене отказавшей подсистемы (устрой
ства, процессора) за счет имеющихся в системе резервов,
либо в ее отключении. В последнем случае необходимо осу
ществить перераспределение задач между оставшимися под
системами, что связано с понижением производительнос
ти системы. После реконфигурации производится восста
новление информации (S_{10}).

Для этого по ходу вычислительного процесса предусмотрены контрольные точки, в которых состояние системы и вычислительного процесса подвергается контролю. В случае положительного результата состояние данной программы (данного процесса), промежуточные результаты, содержание регистров и др. записываются в дополнительной оперативной памяти другого процессора, либо на магнитных лентах или дисках и используются для восстановления информации. В ходе восстановления информации содержание этих дублирующих записей перезаписывается в тот процессор, который после реконфигурации берет на себя функции отказавшего. Затем, начиная с контрольной точки, вычислительный процесс возобновляется (S_{11}).

Аналогичные процедуры проводятся в случае, когда ошибка обнаружена программными средствами. Однако при этом повторение операций не имеет смысла, поскольку программные средства обнаруживают ошибку с опозданием, и поэтому операнды практически не сохраняются.

После обнаружения ошибки программными средствами могут быть использованы тесты (S_{12}). Если тесты подтверждают наличие устойчивого отказа (S_{10}), то производится реконфигурация (S_9), возврат к контрольной точке, восстановление данных (S_{10}) и повторение вычислений (S_{11}). Если устойчивого отказа нет (S_{14}), то повторяются перечисленные операции без реконфигурации. Отметим, что восстановление может оказаться безуспешным также при наличии ошибки в программах, при разрушении информации в контрольных точках и при исчерпании резервов или уменьшении производительности системы из-за отказов ниже допустимого уровня. В этих случаях система переходит в отказовое состояние (S_4).

На рис.28 вершины графа соответствуют отдельным событиям, а дуги графа — переходам от одного события к другому. Предполагается, что после событий S_2, S_3, S_5 и событий, непосредственно следующих за ними, возможен переход к событию (S_4). Соответствующие дуги не показаны в целях упрощения графа. Пусть дуги (рис. 5) имеют

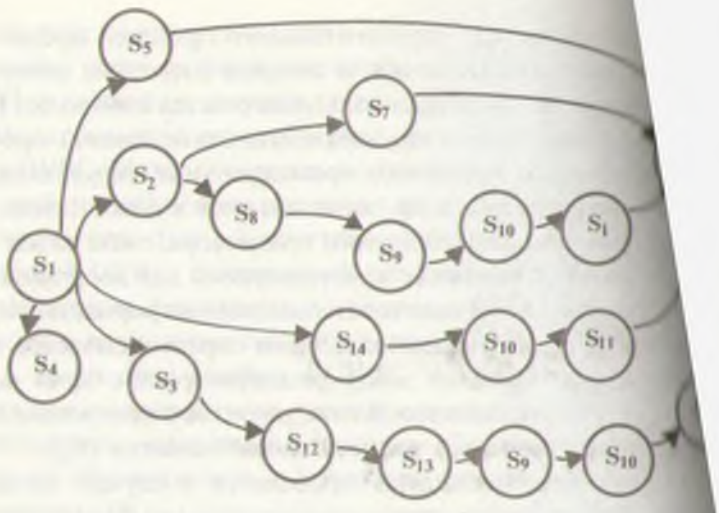


Рис. 28. Граф состояний и переходов процесса восстановления ОУВС

веса, определяющие время между следующими друг за другом событиями ОУВС. Пусть для каждой из дуг графа заданы математическое ожидание и дисперсия времени.

Необходимо определить функцию распределения времени восстановления ОУВС с точностью до первых двух моментов, вероятность восстановления ОУВС автоматическими средствами и среднее время автоматического восстановления ОУВС.

Определим значение коэффициента q разряжения потока сбоев и отказов аппаратуры ОУВС, как условную вероятность того, что система не будет автоматически восстановлена в случае возникновения отказа или сбоя в работе одной из подсистем. Тогда по формуле полной вероятности

$$q = q_т * (1 - q_м) + (1 - q_т) * (q_н + q_р)$$

или после преобразования

$$q = q_1 * (1 - q_н - q_р - q_м) + q_н + q_р.$$

вероятность того, что система не будет восстановлена автоматически за допустимое время $\tau_{доп}$; $q_м$ – вероятность того, что отказ системы маскируется, т.е. перекрывается областями пассивной отказоустойчивости; $q_н$ – вероятность того, что отказ не обнаруживается; $q_р$ – вероятность того, что информация во всех контрольных точках разрушена и вычислительный процесс не может быть восстановлен. Обозначая

$$q_о = q_н + q_р + q_м,$$

имеем

$$q = q_т * (1 - q_о) + q_о - q_м.$$

Чтобы вычислить значения $q_т$, найдем распределение вероятности времени восстановления, которое определяется по дугам графа, соединяющего вершины S_1, S_2, S_6 и S_1, S_3, S_6.

По методу моментов для последовательных дуг графа дисперсии и математические ожидания составляющих времени прохождения графа складываются. Для параллельных дуг графа функция распределения времени восстановления имеет вид смеси. Математическое ожидание смеси

$$m = \sum \pi_j * m_j,$$

где $П_j$ – вероятность j-й составляющей смеси; m_j – математическое ожидание j-й составляющей смеси.

Исходя из известной зависимости $\alpha = \sigma^2 + m^2$, где α – начальный момент второго порядка, а σ – среднее квадратическое отклонение, дисперсия смеси определяется по формуле

$$\sigma^2 = \sum_j П_j(\sigma_j^2 + m_j^2) - m^2,$$

так как

$$\alpha = \sum_j \Pi_j * \alpha_j = \sum_j \Pi_j (\sigma_j^2 + m_j^2),$$

где σ^2 – среднее квадратическое отклонение j-й составляющей смеси.

В частности, в случае двух составляющих имеем

$$\sigma^2 = \pi_1 * \sigma_1^2 + \pi_2 * \sigma_2^2 + \pi_1 * \pi_2 * (m_1 - m_2)^2.$$

Моменты распределения времени прохождения графа находятся путем последовательного его упрощения.

После каждого отказа (сбоя) элемента системы работоспособность ОУВС восстанавливается за некоторое случайное время τ, характеризуемое функцией распределения $F(\tau)$. Вероятность того, что последствия отказа не будут автоматически устранены за допустимое, с точки зрения пользователя, время τ_{don} выражается как $q_\tau = (1 - F(\tau_{don})$

Функция распределения времени восстановления $F(\tau_{don})$, как показано выше, известна только с точностью до первых двух моментов. Поэтому для вычисления $q\tau$ допускается, что функция распределений $F(\tau)$ известна, например $F(\tau)$ – гамма-распределение или нормальное распределение. Поскольку фактическое распределение τ отличается от известных простых законов распределения, то полученная таким образом оценка является приближенной

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Представьте в аналитическом виде сложность средств реконфигурации ОУВС.

2. Какие составляющие являются определяющими при вычислении времени реконфигурации ОУВС?

3. Оцените производительность ОУВС с сетевой организацией ($n = 2$).

4. Какие способы восстановления учитываются при построении модели процесса автоматического восстановления?

4. Определите процедуры восстановления при обнаружении аппаратными средствами ОУВС устойчивого отказа.

5. Постройте граф состояний и переходов процесса восстановления ОУВС.

6. При какой организации работы ОУВС каждый процессор должен иметь таблицу ресурсов?

Глава 7. ПРИМЕРЫ ОРГАНИЗАЦИИ ОТКАЗОУСТОЙЧИВЫХ СИСТЕМ

В настоящем разделе приводятся примеры организ ОУВС, имеющих различное целевое назначение, со ных различными фирмами и в силу этого обладающих ципиальными отличиями в реализации средств обесп ния отказоустойчивости. Большинство современных ОУ являются микропроцессорными, т.е. содержат микроп цессоры и микросхемы микропроцессорных комплекто Характерной особенностью таких систем является испол зование, наряду с микропроцессорами, других больш интегральных схем, которые позволяют расширить фун циональные возможности и повысить производительност отказоустойчивых систем.

Пассивно отказоустойчивая самопроверяемая систем «Stratus» использует дублирование с контролем-сравнени ем, причем все основные функции выполняются четыре ды. Во-первых, каждая подсистема (ТЭЗ) дублирована. Во вторых, каждая подсистема включает пару идентичных схем с идентичными входными сигналами в целях контроля. Если только выходы этих схем будут отличаться, сравнивающие схемы вырабатывают сигнал об ошибке. При нормальной работе дублирующие системы работают строго синхронно. Если только в одной из них возникает ошибка, то эта система отключается, а другая — продолжает работать.

Система SIFT (программное обеспечение устойчивости к отказам) представляет собой вычислительную систему, предназначенную для управления полетом самолета в особо сложных условиях. Основным принципом, заложенным в СОО, является параллельное выполнение каждой программы несколькими блоками обработки данных. В качестве блоков обработки данных (БОД) и устройств сопряжения с периферийным оборудованием системы используются стандартные мини-ЭВМ. Процедуры обнару-

[illegible] и анализа ошибок и реконфигурации системы возлагается на программное обеспечение, отсюда и название [illegible]. Локализация отказов достигается применением [illegible] разработанной избыточной системы шинных соединений блоков обработки данных. Как минимум трехкратное резервирование выполнения программ позволяет [illegible] влияние любого одиночного отказа БОД или [illegible], а устойчивость к последовательности отказов со[illegible] за счет реконфигурации системы. При этом одинаковые программы выполняются БОД независимо, что позволило отказаться от жесткой синхронизации процессоров.

Система STAR (самопроверяемая и ремонтируемая вычислительная машина) [17] предназначена для беспилотных космических полетов большой продолжительности (до [illegible] лет). С учетом ограничений на потребляемую мощность, вес и габариты аппаратуры выбран следующий принцип организации СОО. В каждый момент времени функционирует одна вычислительная машина, снабженная эффективными схемами контроля для обнаружения неисправностей, и достаточным количеством резервных блоков. При этом используется ненагруженный резерв, т.е. на резервные блоки не подается напряжение питания. Поскольку предусмотрена только одна рабочая вычислительная машина, потребовалось организовать специальное аппаратное «ядро», которое обеспечило бы диагностику отказов в вычислительной машине, автоматическую замену неисправных блоков на резервные и выработку управляющих сигналов, запускающих программную процедуру восстановления. Это «ядро», получившее название ПКВ (процессор контроля и восстановления), представляет собой аппаратный блок с гибридной избыточностью (мажоритарное резервирование плюс ненагруженный резерв).

Таким образом, в системе SIFT большая часть функций СОО реализуется в виде программного обеспечения, а в системе STAR – виде аппаратного блока. Но тем не менее, и в том, и в другом случае необходимо большое количество избыточной аппаратуры и специальное программное обеспечение. Следует обратить внимание пользователя также на то, что эти два примера приводятся здесь как иллюстрация двух разных крайних подходов к реализации

СОО для конкретных применений, и по существу
хорошо просматривается та совокупность задач, с к
ми сталкивается разработчик современных ОУВС.

7.1. ОТКАЗОУСТОЙЧИВАЯ СИСТЕМА ФИРМЫ STR

Отказоустойчивая система фирмы STRATUS со
следующие устройства (рис. 29): центральный проц

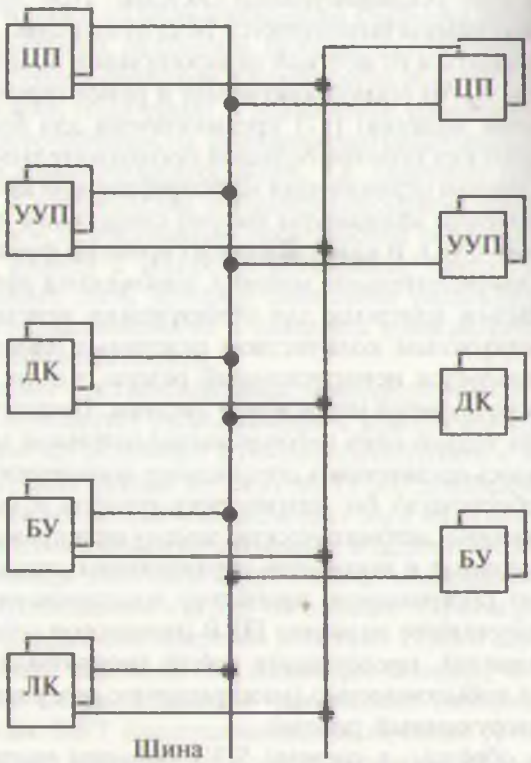


Рис. 29. Отказоустойчивая система фирмы STRATUS

(ЦП), устройство управления памятью (УУП), дисковый контроллер (ДК), блок управления (БУ), ленточный контроллер (ЛК) и общую шину. Каждое устройство системы задублировано, и каждое дублированное устройство, в свою очередь, проверяется в процессе функционирования с по-

такого же устройства. Фактически в системе каждое функциональное устройство (ЦП, УУП, ДК, БУ) реализовано в виде двух самопроверяемых блоков, в которых для построения средств встроенного контроля используется метод дублирования.

В случае неисправности какого-либо блока системы производится переключение на исправную пару устройств. Факт неправильной работы фиксируется несовпадением выходных результатов пары устройств, образующих самопроверяемый блок. Поскольку общая шина также задублирована, то при отказе одной из них производится переключение на исправную шину. Таким образом, полная отказоустойчивость в системе STRATUS достигается за счет значительных аппаратурных затрат, но при этом гарантируется высокая полнота обнаружения неисправностей (за счет контроля дублированием).

Интересной особенностью рассматриваемой системы является исключение временных затрат для восстановления по сбоям. При сравнении выходных результатов на какой-либо паре устройств эта пара исключается из рабочей конфигурации и подключается исправная пара. Затем с помощью тестовых средств определяется вид неисправности (устойчивая или неустойчивая) и в случае устойчивой неисправности производится замена неисправного устройства пары. Поскольку восстановление в системе выполняется на уровне аппаратуры, то никакие программные способы образования контрольных точек не используются. Кроме того, в системе не предусмотрена возможность рассылки между блоками информации об их состоянии. Как следствие, все это существенно упростило систему.

7.2. СИСТЕМА SIFT

Первый вопрос, который решался при выборе архитектуры ОУВС SIFT, был вопрос о видах отказов, к которым необходимо обеспечить устойчивость. При этом главным фактором, повлекшим за собой принятие решения,

был тип БИС элементной базы, применяемой для построения системы. В связи с этим разработчиков не удовлетворял традиционный подход, в котором рассматривались отказы типа тождественный 0 или 1 на входах и выходах отдельных элементов. В современных устройствах, построенных на элементах БИС, неисправности компонентов могут вызывать сложные последствия, которые практически трудно предсказать. Поэтому целесообразно не выделять виды отказа, а различать лишь исправные и неисправные блоки. В соответствии с этим система обнаружения ошибок строится с учетом только выявления искаженных данных, без возможности последующего выяснения их конкретных причин.

Следует отметить однако, что при отказах устройств сопряжения необходимо детально исследовать их последствия для работы сопрягаемых блоков. Такое решение вопроса о видах отказов в сильной степени повлияло на выбор программного способа реализации функций обнаружения ошибок, исправления их, локализации отказов и реконфигурации системы.

На рис. 30 представлена структурная схема SIFT. Вычисления в ней выполняются основными процессорами. Результаты вычислений каждого процессора запоминаются в основном в запоминающем устройстве (ЗУ), связанном только с одним процессором. Процессор и его ЗУ соединены обычным широкополосным каналом связи. Структура процессоров и ЗУ ввода-вывода аналогична структуре основных процессоров и ЗУ, с той лишь разницей, что их вычислительная мощность и емкость памяти значительно меньше. Периферийные устройства подключены к входным и выходным блокам системы, которыми в данном случае являются датчики и управляющие органы самолета. Каждый процессор и связанное с ним ЗУ образуют блок обработки данных, и каждый такой блок подключен к многомашинной системе связи.

Система SIFT выполняет ряд заданий, каждое из которых представляет собой последовательность итераций. Входными данными для выполнения очередной итерации не-

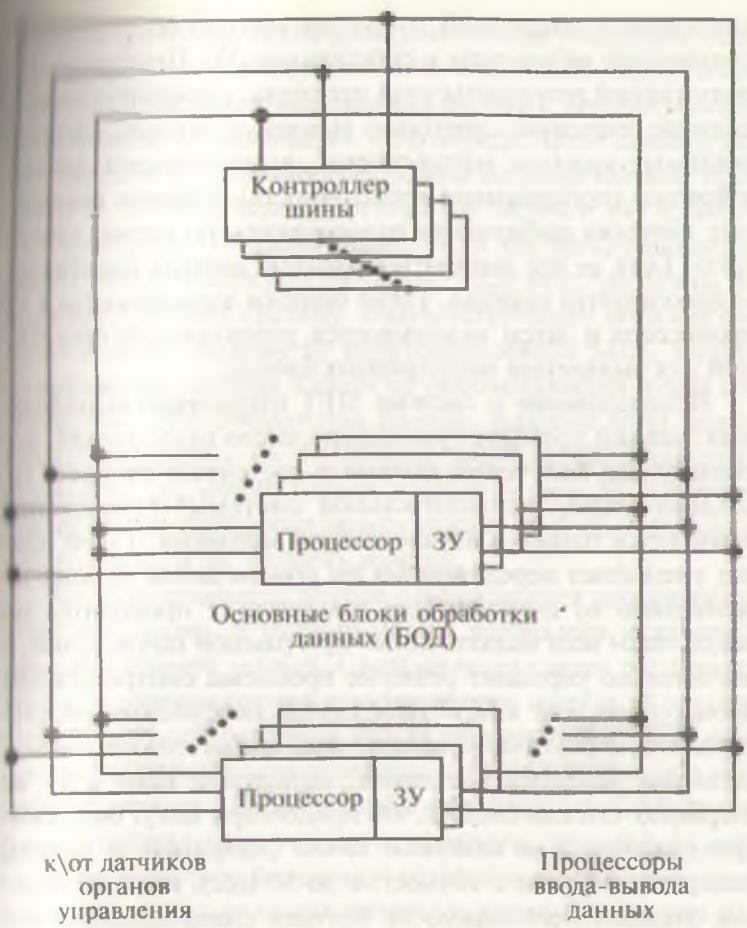


Рис. 30. Структурная схема системы SIFT

которого задания являются полученные на предыдущей итерации выходные данные некоторой совокупности заданий (включающей в себя и рассматриваемое задание). Входные и выходные сигналы всей системы формируются при выполнении заданий процессорами ввода-вывода.

Отказоустойчивость системы обеспечивается за счет независимого выполнения каждой итерации любого задания несколькими блоками (аппаратная избыточность). После

выполнения очередной итерации процессор пересылает полученные результаты в собственное ЗУ. Процессор, использующий результаты этой итерации, определяет их правильные значения, сравнивая выходные данные, сформированные каждым процессором, выполнявшим данную итерацию (программная избыточность). Обычно правильные значения выбираются голосованием по методу «два из трех». Если не все варианты выходных данных идентичны, то фиксируется ошибка. Такие ошибки записываются в ЗУ процессора и затем используются управляющей программой для выявления неисправных блоков.

Использование в системе SIFT итеративно выполняемых заданий позволяет уменьшить число голосований, поскольку для получения данных о состоянии самолета (и, следовательно, вычислительной системы) голосование проводится только в начале каждой итерации. Такой подход уменьшает передаваемый по шинам поток данных по сравнению со схемами, где голосование проводится по результатам всех выполненных программой вычислений, и значительно упрощает решение проблемы синхронизации процессоров, так как в этом случае необходимо обеспечить, чтобы различные процессоры, назначенные для выполнения некоторого задания, выполняли одну и ту же итерацию. Отсюда следует, что процессоры могут быть синхронизированы не слишком точно (например, в данном конкретном случае с точностью до 50 мкс), и таким образом отпадает необходимость жесткой синхронизации команд и тактовых импульсов. Одно из преимуществ «слабой» синхронизации состоит в том, что любая итерация задания может быть запланирована для выполнения разными процессорами в различные, но близкие друг к другу моменты времени. Поэтому маловероятно, чтобы одновременно происшедшие сбои в нескольких процессорах привели к одинаковым ошибкам при выполнении одного задания.

Число процессоров, выполняющих одно и то же задание, может изменяться в зависимости от типа задания и

момента его выполнения (например, если некоторое задание, некритичное в один момент, становится критичным в другой). Распределение заданий по БОД, вообще говоря, различно для каждого блока. Оно определяется динамическим образом так называемой общей исполнительной программой, которая ведет диагностику ошибок для выявления неисправных блоков и шин. Если эта программа определила, что в некотором БОД возникла неисправность, то она перестраивает систему, соответствующим образом перераспределяя задание по работоспособным БОД. Защита от искажений данных в системе обеспечивается путем соответствующего соединения блоков. Каждый БОД может считывать данные из ЗУ любого модуля, но записывать результаты он может только в собственное ЗУ. Поэтому неисправный процессор может исказить данные только в собственной, а не «чужой» памяти.

Все неисправности внутри одного модуля рассматриваются одинаково, а именно — как источники искажения данных в памяти модуля. Система не пытается распознать причину возникновения неисправности модуля. В частности, не делается различия между неисправным ЗУ и процессором, передавшим неверные данные в исправное устройство памяти. Для того чтобы неисправный блок не нарушил правильную работу исправного, каждый из них снабжен автономной системой управления. Однако исправный блок может получить неверные данные из памяти неисправного.

Устранение влияния такой ошибки и происходит за счет того, что исправный блок получает несколько копий данных (как минимум три) из разных ЗУ по разным шинам, а правильную версию данных процессор определяет посредством мажоритарного голосования. После этого исправный процессор определяет по результатам голосования неисправный БОД, а с помощью специальных средств может быть определена и неисправность шины. В первом случае неисправный БОД отключается и его функции передаются

другим исправным БОД, во втором случае передача информации по неисправной шине прекращается, а процессоры будут получать данные по другим шинам. После такой реконфигурации система восстановит способность противостоять новому отказу, если, конечно, осталось достаточное число исправных блоков и шин.

7.3. СИСТЕМА STAR

Как было отмечено выше, основными факторами, определившими выбор способа реализации СОО в данной ОУВС, были жесткие ограничения на ее физические параметры. Эти же факторы определили и выбор элементной базы – биполярные интегральные схемы малого и среднего уровня интеграции. Применение таких элементов потребовало тщательной проработки задачи разбивки ВС на модули замены. Например, центральный процессор состоит более, чем из 1000 кристаллов, что дает значительную суммарную интенсивность отказов. Чтобы обеспечить заданную надежность системы, оказалось необходимым разделить процессор на четыре модуля и каждый из них обеспечить определенным числом резервных модулей.

В системе STAR предусмотрен иерархический принцип обнаружения и устранения влияния неисправностей, восстановления и реконфигурации системы. В частности, модули высокого уровня осуществляют проверку и изоляцию шин связи и модулей нижнего уровня, управляют перегрузкой памяти и реконфигурацией при замене неисправных модулей, производят инициализацию при перемежающихся отказах. Таким образом создают условия функционирования этой системы с планируемым сроком работы в течение пяти лет.

На рис. 31 представлена структурная схема системы STAR, которая состоит из семи типов различных модулей, соединенных между собой двумя четырехпроводными шинами и имеющих резервные копии. На рисунке представлены: УП – управляющий процессор, который содержит

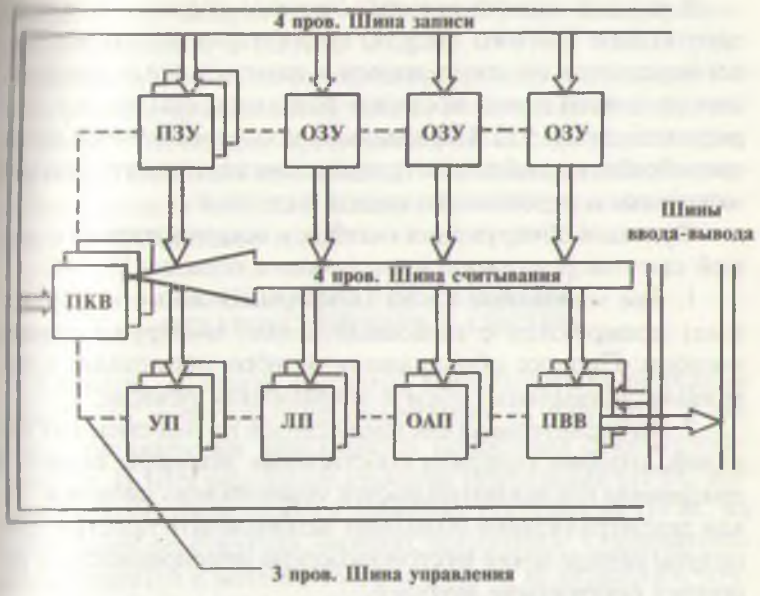


Рис. 31. Структурная схема системы STAR

счетчик адресов и индексные регистры, а также осуществляет модификацию адресов команд перед их выполнением; ЛП – логический процессор, выполняющий логические операции над информационными словами (напряжение питания подается сразу на две копии); ОАП – основной арифметический процессор для арифметических операций над информационными словами; ПЗУ –постоянное запоминающее устройство емкостью 4 К слов; ОЗУ – оперативное запоминающее устройство емкостью 4 К слов (питание подается по крайней мере на две копии, и непосредственную адресацию имеют 12 блоков), ПВВ – процессор ввода-вывода, содержащий буферные регистры ввода-вывода, и процессор прерывания, управляющий запросами на прерывание; ПКВ – процессор контроля и восстановления, который управляет работой вычислительной системы и осуществляет восстановление (питание подается одновременно на три копии).

В каждый момент времени питание подается только на одну копию каждого модуля, информация между модулями передается по шине записи в память и шине считывания из памяти в виде восьми 4-разрядных ссылок, т.е. разрядность слова – 32. Параллельно-последовательный принцип обработки выбран по причине снижения потребляемой мощности и вероятности отказа в системе.

Функции обнаружения ошибок и восстановления в данной системе реализуются следующим образом [5, 17]:

1. Все машинные слова (информационные и командные) кодируются с помощью кодов, обнаруживающих ошибки. Процесс обнаружения ошибок происходит с помощью специальных схем в оперативном режиме.

2. Вычислительная система делится на ряд сменных модулей, которые содержат собственные декодеры команд и генераторы последовательностей управляющих сигналов. Такая децентрализация позволяет использовать простые процедуры определения местоположения неисправности и упрощает сопряжение модулей.

3. Обнаружение неисправностей и восстановление выполняются с помощью специальной аппаратуры. В случае отказа ЗУ на помощь аппаратуре, служащей для восстановления, приходит программное обеспечение.

4. Производится распознавание сбоев и их влияние корректируется посредством повторения сегмента текущей программы. В случае отказа модуля он заменяется резервным.

5. Замена осуществляется путем переключения питания – оно снижается с неисправных модулей и подается на резервные. Информационные линии всех модулей постоянно подсоединены к шинам через изолирующие схемы. Модули, на которые питание не подано, не оказывают влияния на шины.

Модуль ПКВ защищен посредством тройного нагруженного резерва.

Модуль ПКВ – самый оригинальный модуль в этой вычислительной системе. Он следит за работой шин посредством проверки справедливости кодов с обнаружением ошибок, а также за сообщениями о состоянии различных функциональных модулей. Если приходит сигнал ошибки

какого-либо модуля или в шину поступает закодированная информация с выхода модуля, то ПКВ инициирует повторное выполнение сегмента программы и, если ошибка повторяется, производится замена неисправного блока резервным с помощью трехпроводной шины управления. Система STAR одна из первых ОУВС и идеология ее построения оказала большое влияние на последующие разработки ОУВС данного типа.

7.4. ПЕРСПЕКТИВА РАЗВИТИЯ ОТКАЗОУСТОЙЧИВЫХ СИСТЕМ

Сегодня, с одной стороны, происходит унификация средств ВС, с другой – специализация их функций. В эти рамки органически вписывается концепция отказоустойчивости. И хотя отдельные способы реализации ОУВС постоянно совершенствуются, принцип построения ОУВС на ближайшие десятилетия, видимо, сохранится. Следовательно, сохранятся и математические модели их описания, точность и адекватность которых будет повышаться по мере реализации ОУВС.

Быстрое развитие элементной базы ВС позволяет разработчику использовать элементы с малой потребляемой мощностью, повышенной плотностью упаковки, низкой стоимостью, а это приводит к изменению характера целевой функции при проектировании ОУВС. Уже нет необходимости обеспечивать устойчивость к отказам при минимальном количестве дополнительной аппаратуры. Напротив, для лучшего обнаружения неисправностей и восстановления можно задействовать большее количество аппаратуры. При этом оказывается эффективным использовать простейшие методы введения избыточности.

Например, имея стандартный однокристальный микропроцессор, можно для обнаружения неисправностей ввести в систему два процессора, работающих синхронно, и сравнивать их выходы. Если один из процессоров выходит из строя, то эта пара процессоров заменяется исправной резервной парой. Это оказывается экономически выгоднее, чем проектирование нового процессора с оптимальными встроенными схемами контроля.

Отталкиваясь от идеи сменных модулей системы STAR в современных ОУВС, целесообразно уровень разбиения на модули производить не ниже микро-ЭВМ. Особенно это важно для распределенной сети вычислительных машин. В таких системах оказывается эффективнее алгоритмы восстановления закладывать в отдельные ЭВМ, а не в специальный модуль, как это было в системе STAR.

Для вычислительных систем, проектируемых на основе многокристальных процессоров, наиболее актуальной становится задача разработки самопроверяемых вычислительных модулей с встроенной аппаратурой для обнаружения своих собственных неисправностей (включая и аппаратуру контроля). Разработка таких модулей приводит к созданию стандартного набора самопроверяемых сверхбольших интегральных схем. Такой набор позволит значительно упростить процесс проектирования ОУВС и реализацию основных функций средств обеспечения отказоустойчивости.

Единая целевая комплексная программа использования микропроцессоров предусматривает создание, освоение и производстве и ввод в эксплуатацию систем и комплексов на базе микропроцессорных средств, в частности:

– автоматизированных технологических комплексов;

– систем и устройств автоматического управления и регулирования;

– информационных систем предприятий, гибких переналаживаемых производств и систем управления для них;

– систем для научных исследований, проектно-конструкторских работ; обучения и отладки микропроцессорной техники;

– измерительных систем, комплексов и приборов.

Обеспечение отказоустойчивости таких систем тесно связано с развитием системных принципов проектирования, совершенствованием технологии изготовления компонентов системы и расширением круга задач. Большое разнообразие требований, предъявляемых к функциональным возможностям, приводит к различным архитектурам отказоустойчивых систем.

Можно выделить следующие основные факторы, пре-

определяющие дальнейшее развитие архитектур отказоустойчивых систем:

1. Дальнейшее возрастающее использование большого количества (до 10^{-3}—19^{6}) БИС и СБИС в различных компонентах систем.

2. Развитие принципов организации физических и программных связей между элементами системы.

3. Совершенствование структурной организации системы, обеспечивающей высокую степень параллельности обработки данных.

4. Новые подходы к построению математического обеспечения отказоустойчивых систем, направленных, в частности, на повышение их помехозащищенности, отказоустойчивости, устранение влияния различных видов ошибок.

5. Формирование нового взгляда на обслуживание систем, оценку их эффективности, надежности и достоверности функционирования.

Применение БИС и СБИС диктует определенные требования к архитектуре системы. В настоящее время большое внимание при построении БИС и СБИС уделяется вопросам их диагностируемости и самоконтролируемости. Высокая аппаратурная избыточность СБИС позволяет значительно повысить время наработки на отказ отдельных компонентов системы.

В ближайшие годы ожидается увеличение емкости оперативной памяти до 16 Мбит и выше, а микропроцессоры будут содержать до 10 и более вентилей на 1 мм^2 [30, 31].

Наиболее перспективным является использование ЗУ на полупроводниковых СБИС, магнитных доменах, ЗУ с зарядной связью и лазерной памятью. В частности, использование лазерной памяти позволяет получить ЗУ объемом в сотни Мегабит при времени доступа 10—20 мкс.

Широкое распространение получают 16- и 32 - разрядные микро-ЭВМ, что приводит к дальнейшему росту функциональных возможностей микропроцессорных систем, росту их производительности.

Применение способов повышения глубины диагностирования СБИС, их контролепригодности (например, распространенный подход, состоящий в перестройке струк-

туры БИС ЗУ в длинный сдвигающий регистр) обеспечивает своевременное выявление неисправностей в отдельных компонентах системы при отбраковочном контроле элементов и в процессе диагностирования при эксплуатации.

Можно ожидать, что по мере дальнейшего развития подсистем связи распределенных систем для повышения пропускной способности каналов связи будет широко применяться разделение по частоте и по времени передаваемых сообщений, что обеспечит высокую пропускную способность этих каналов. Это потребует проведения дальнейших работ по освоению пакетной передачи данных, совершенствования протоколов обмена, организации взаимодействия новой системы с уже существующими.

Программное обеспечение сейчас занимает по удельному весу 50 — 90 % общей стоимости системы. Поэтому одной из основных задач можно считать разработку подходов к автоматизации программирования, средств выявления ошибок в ПО. Унификация и упрощение ПО за счет использования структурного программирования в сочетании с повышением отказоустойчивости ПО позволит создавать эффективные системы с высокой преемственностью математического обеспечения.

Большое значение при эксплуатации отказоустойчивых систем имеют протоколы обмена данными между элементами системы. Многие из существующих протоколов обмена ориентированы на поблочную передачу данных. Блоки обычно включают достаточно большое число разрядов контроля, что затрудняет их обработку при приеме. При некоторых реализациях систем, ориентированных, в частности, на непрерывную передачу данных, более оправданным является применение избыточных межблочных кодов. Поэтому со временем большое распространение получат протоколы обмена, в которых будет применяться избыточное кодирование передаваемых потоков данных.

Быстрый рост объемов выпуска микропроцессоров, БИС и СБИС, создание многомашинных систем, построенных на их основе, ставит задачу эффективного обслуживания таких систем. Круг пользователей многомашинных систем, различающихся уровнем подготовки, расширяется. Вместе

с тем превышение темпов роста количества микропроцессорных устройств и систем по сравнению с темпами роста обслуживающего их квалифицированного персонала требует разработки единых принципов обеспечения их контролепригодности с учетом рационального использования не только самих технических средств, но и затрат, связанных с их обслуживанием. Фактор обслуживания становится одним из доминирующих факторов обеспечения эффективной работы систем.

Таким образом, комплекс перечисленных факторов, а также дальнейшее развитие единых принципов обеспечения устойчивости устройств и систем к возникающим отказам позволит создавать высокопроизводительные отказоустойчивые системы с длительным временем автономного функционирования.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте характеристику особенностям микропроцессорных ОУВС.
2. Опишите процедуру восстановления вычислительного процесса в пассивно отказоустойчивой ВС (Stratus).
3. Какой вид контроля используется в системе ВС Stratus?
4. Установите, какие дополнительные функции необходимо вводить для восстановления ошибочного результата в системе SIFT?
5. Укажите способ реализации функций обнаружения ошибок, исправления их, локализации отказов и конфигурации в системе SIFT.
6. Для каких целей предназначена ОУВС STAR?
7. Какова функция процессора ПКВ в системе STAR?
8. Какие основные факторы, предопределяющие дальнейшее развитие архитектур отказоустойчивых систем можно выделить?

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Авижиенис А.А. Отказоустойчивость – свойство, обеспечивающее постоянную работоспособность цифровых систем // ТИИЭР, 1978. Т.66. № 10.
2. Расулова С.С. Методы оценки надежности и обеспечение отказоустойчивости вычислительных систем. Препринт. НПО «Кибернетика». Т., 1991.
3. Коваленко А.Б., Гула В.В. Отказоустойчивые микропроцессорные системы. Киев.: Техника, 1986.
4. Расулова С.С. Надежность вычислительных машин и систем. Учебное пособие. Часть I. Т.: ТашГТУ, 1995.
5. Афонин В.А., Ладыгин И.И. Построение отказоустойчивых вычислительных систем. М.: МЭИ, 1990.
6. Расулова С.С., Хайдаров Ш.А. Организация восстановления в микропроцессорных отказоустойчивых системах // Вопросы технической диагностики: Сб. науч. тр. Ростов-на-Дону: Рост.инж.-строит. ин-т, 1990.
7. Бекмуратов Т.Ф., Расулова С.С., Икрамов С.А., Хайдаров Ш.А. Анализ и оценка надежности восстанавливаемых вычислительных комплексов на базе мини-ЭВМ //Изв. АН УзССР. СТН. 1990. Вып.3.
8. Иыуду К.А. Надежность, контроль и диагностика вычислительных машин и систем. М.: Высшая школа, 1989.
9. Schooman M.L. Architecture of Fault-Tolerant Computers.// Computer, 1996, № 3.
10. Расулова С.С., Гаибназаров С.Д. Контроль и диагностика вычислительных и микропроцессорных систем: Учебное пособие. Т.: ТашГТУ, 1995.

11. Расулова С.С. Структурные методы обеспечения надежности вычислительных систем // Тез. докл. IX Всесоюз. совещ. по техн. диагностике. Ростов-на-Дону, 1991.
12. Расулова С.С. Влияние диагностического обеспечения на надежность микро-ЭВМ // Тез. докл. 7-го Всесоюз. совещ. по техн. диагностике и отказоустойчивости. М., 1990.
13. Каган Б.М. Электронные вычислительные машины и системы. М.: Энергоатомиздат, 1991.
14. Иыуду К.А., Кривошенков С.А. Математические модели отказоустойчивых вычислительных систем. М.: МАИ, 1994.
15. Согомонян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989.
16. Расулова С.С. Автоматизированный расчет характеристик надежности реконфигурируемых систем. // Труды всесоюз. школы «Автоматизация создания математического обеспечения и архитектуры систем реального времени». Иркутск, 1990.
17. Головкин Б.А. Параллельные вычислительные машины. М.: Наука, 1994.
18. Расулова С.С., Хайдаров Ш.А. Использование цепей Маркова для анализа надежности микро-ЭВМ // Вопросы кибернетики. Т.: РИСО АН УзССР. 1989. Вып. 139.
19. Расулова С.С., Хайдаров Ш.А. Прогнозирование и оптимизация надежности многопроцессорной вычислительной системы методом моделирования // Докл. АН, 1990, вып. 3.
20. Икрамов С.А., Иргашев Ф.А., Расулова С.С. Моделирование и оценка надежности отказоустойчивых вычислительных систем // Тр. Междунар. конф. «Диагностич. обеспеч. цифр. систем». ЧССР. Брно, 1986.
21. Расулова С.С. Выбор математической модели для

оценки параметров надежности отказоустойчивых вычислительных систем: В кн. «Проблемы создания и использования мини-микро-ЭВМ». Вильнюс, 1981
22. Расулова С.С. Учет характеристик средств диагностирования при оценке надежности ПЭВМ// Сб. тр. науч-техн. семинара «Применение ПЭВМ в системах проект., контроля и диагн. РЭА». М., 1989.
23. Расулова С.С. Алгоритм и программа расчета надежности вычислительных систем с реконфигурацией // Тез. докл. Всесоюз. науч. конф. «Эконом. проблемы страны». Т., 1990.
24. Каган Б.М., Мкртумян И.Б. Основы эксплуатации ЭВМ. М.: Энергоатомиздат, 1992.
25. Расулова С.С., Сафарова Н. Численный метод нахождения вероятности безотказной работы высоконадежных систем // Тр. 2-ой Всесоюз. научн.-техн. конф. «Живучесть и реконфигурация информ.-вычисл. систем». Киев, 1989. Вып. 2.
26. Расулова С.С. Особенности контроля и диагностирования микропроцессорных систем. М.: Деп. в ЦНИИТЭИС, 1991. № 3671.
27. Расулова С.С., Хайдаров Ш.А. Программа оценки, моделирования и прогнозирования надежностных характеристик управляющих вычислительных комплексов // Всесоюз. ФАП. Инв. № 5090000113. М.: ВНИТИ-центр, 1990.
28. Расулова С.С., Хайдаров Ш.А. Программа оптимизации надежности многопроцессорной ВС с динамическим резервированием// ГОС ФАП СССР № 5090000949. М., 1991.
29. Расулова С.С., Гаибназаров С.Д. Применение методов расчета и обеспечение надежности микропроцессорных систем в профессионально-техническом образовании// Проблемы становления и функционирования образовательных комплексов в системе профессионального образования. Т.: «Фан», 1996.

30. Triveoli A.K., Schooman M.L., A many state Markov Model for the Estimation and Prediction // Proc. of the Internationation Conference on Reliable Software, 1996.
31. Додонов А.Г. и др. Введение в теорию живучести вычислительных систем. Киев.: Наукова думка, 1990.
32. Березюк Н.Т., Гапунин А.Я., Подлесный Н.И. Живучесть микропроцессорных систем управления. Киев.: Техника, 1998.

ОГЛАВЛЕНИЕ

САЙЯРА САМАТОВНА РАСУЛОВА
АБДУРАШИД АБДУВАХИДОВИЧ РАШИДОВ

ПОСТРОЕНИЕ ОТКАЗОУСТОЙЧИВЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Учебное пособие

Ташкент — «Mehnat» — 2004

Зав. редакцией *А.Бобониязов*
Редактор *Г.Хубларов*
Художественное оформление *Х.Кутлуков*
Техн. редактор *Т.Смирнова*
Корректор *Г.Усманова*

Подписано в печать 14.01.2004. Формат 84x108 $^{1}/_{32}$. Гарнитура «Таймс», бумага офсетная. Уч.-изд. л. 9,0. Усл. п. л. 9,0.
Тираж 1000 экз. Заказ № 3035.
Цена договорная.

Издательство «Mehnat», 700129. г. Ташкент, ул. Навои, 30.
Дог. № 62—2003.

Отпечатано в типографии №1 Узбекского Агентства по печати и информации. г.Ташкент, ул. Сагбан, тупик 1, дом 2.