

САМОДИАГНОСТИРОВАНИЕ ЦИФРОВЫХ СИСТЕМ С РЕКОНФИГУРАЦИЕЙ

В.А. Ведешенков

Институт проблем управления им. В.А. Трапезникова, г. Москва

Предложен метод самодиагностирования компонентов (модулей и линий связи) цифровых систем с реконfigurацией, допускающий устойчивые кратные отказы ограниченного числа компонентов, в основу которого положен принцип расширяющихся областей. Дан пример самодиагностирования отказавших компонентов в 2-отказоустойчивой системе.

ВВЕДЕНИЕ

Многопроцессорные вычислительные системы (МВС) широко применяются в автоматизированных системах управления различными объектами. Одно из обязательных свойств управляющих систем реального времени состоит в их отказоустойчивости, т. е. способности правильно функционировать в течение заданного времени в условиях возникновения отказов и сбоев в работе их компонентов [1, 2]. Для обеспечения необходимой отказоустойчивости МВС часто применяется стратегия динамической избыточности, включающая в себя этапы контроля работоспособности системы, поиска места нахождения отказавшего компонента, реконfigurации системы для его замены исправным компонентом из числа резервных, восстановления функционирования системы. Методы синтеза графа структуры отказоустойчивых систем, предложенные в работе [2], позволяют построить граф с заданной степенью m отказоустойчивости (m -ОУ граф) для целевого (исходного не избыточного) графа достаточно общего вида. Инвариантно-групповой подход к синтезу приводит к получению структур m -ОУ графов регулярного вида, что упрощает процедуры их анализа в задачах диагностирования и реконfigurации. Но работ по диагностированию отказавших компонентов и реконfigurации в отказоустойчивых системах немного [3–5]. В настоящей статье основное внимание уделено разработке метода самодиагностирования (СД) отказавших компонентов

в цифровых системах с реконfigurацией, построенных с помощью алгоритмов синтеза [2].

Методы системного диагностирования отказов в цифровых системах (моделях МВС) основаны на том, что, ограничивая максимальное число одновременно отказавших модулей (процессоров МВС), можно реализовать необходимые диагностические функции с помощью оставшихся исправных модулей. Модели системного диагностирования, максимально абстрагированные от конкретной реализации МВС, позволяют применять общие методы для диагностирования отказов в системах, построенных на основе процессоров и связей различной физической природы. Взаимодействие отдельных частей системы (процессоров, модулей) при реализации диагностических процедур в данной работе описывается с помощью модели Барси – Грандони–Маестрини (сокращенно, модели БГМ) [6]. Она основана на предположении, что каждый модуль, на которые разбивается исходная МВС, может в одиночку проверить любой другой модуль, имеющий с ним непосредственные связи. Для этого первый модуль (проверяющий, в дальнейшем сокращенно МПш) подает на входы проверяемого модуля (сокращенно МПм) последовательность тестовых воздействий (тест), получает ответные реакции МПм на этот тест, сравнивает их с эталонными, а результаты сравнения оценивает в двоичной форме: 0 – результаты правильные, МПм исправен, 1 – результаты неправильные, МПм неисправен. Предполагается: оценка результатов теста всегда достоверна, если исправен МПш; она равна 1, если отказали оба модуля; недостоверна

(0 или 1) только в том случае, когда МПЩ отказал, а МПм — исправен. Предполагается также, что неисправности модулей — устойчивы, а тесты полные, т. е. обнаруживают всевозможные неисправности модулей.

Подчеркнем, что техническое состояние МПм (компонента) достоверно определяется двоичным результатом его тестовой проверки: 0 — исправен, 1 — неисправен, если исправный МПЩ проверял только один модуль (компонент). Если же в данном тесте одновременно проверялись несколько компонентов, то по единичному результату нельзя однозначно указать отказавший компонент, нужны дополнительные проверки (тесты). Выбору необходимых проверок и разработке методов обработки их результатов и посвящены методы системного диагностирования.

В пользу модели БГМ говорит повышенная сложность модулей (процессоров, машин, устройств связи) современных МВС. Поэтому ничтожно мала вероятность того, что при тестировании неисправного МПм также неисправный МПЩ выдаст оценку 0: МПм исправен. Другими словами, нулевой результат, выдаваемый МПЩ после проверки МПм, с большой вероятностью подтверждает исправность МПм. Такая предпосылка существенно упрощает процедуру диагностирования по сравнению с моделью ПМЧ [7], так как для идентификации исправных модулей достаточно 0-путей единичной длины (в терминологии работы [8]). Правомочность модели БГМ подтверждают и результаты двухлетних экспериментов с алгоритмом распределенного диагностирования Robust [9], показавших, что отказавший модуль очень редко выдает какие-либо данные. Значительно чаще он прекращает выполнение диагностического алгоритма, не выдает никакой информации и потому идентифицируется исправным МПЩ как отказавший.

Суммарный результат проверки модулей системы образует двоичный вектор R , который называется реальным синдромом [6, 7]. Поскольку далее тестовая информация будет представляться в табличном виде, то будем говорить об обработке тестовой информации, а не о дешифрации реального синдрома. Задача обработки тестовой информации заключается в том, чтобы по ней и известной структуре тестовых связей цифровой системы определить подмножество отказавших компонентов (модулей, связей). В большинстве алгоритмов системного диагностирования предполагается (или констатируется), что результаты тестов собираются и обрабатываются внешним устройством.

Другой подход основан на СД состояния компонентов системы путем реализации необходимых алгоритмов, включая сбор и обработку результатов тестирования, на исправных процессорах самой системы (см. например, работы [9–12]). Общие

черты большинства подобных алгоритмов, получивших также название распределенных, заключаются в использовании модели ПМЧ [7], наличии полносвязного диагностического графа, исправности линий связи. Последняя предпосылка частично объясняется тем, что в системах, где реализованы распределенные алгоритмы [11, 12], отказы магистральных линий связи между модулями плохо диагностируются логическими методами.

Ниже предлагается метод СД цифровых систем с реконфигурацией, в основу которого положен принцип расширяющихся областей. Выполнение необходимых процедур начинается с исправного модуля и последовательно передается другим модулям, исправность которых подтверждена предшествующими проверками. Начальный модуль процесса СД определяется по результатам работы системы контроля [13].

Реализация того или иного способа диагностирования зависит от структуры тестовых связей системы, числа тестов, проверяющих каждый модуль, а также количества одновременно отказавших компонентов, топологии их расположения. Максимальное число одновременно отказавших компонентов, однозначно диагностируемых по результатам тестирования системы, называется степенью диагностируемости t системы. Большинство оценок значения t получено в предположении, что отказывают только модули, а связи исправны. Так, в работе [6] для модели БГМ доказано, что в однократно диагностируемых системах из n модулей, обладающих полносвязным диагностическим графом, для степени диагностируемости t справедлива оценка $t \leq (n - 2)$. Предположение о возможности одновременного отказа модулей и связей существенно затрудняет получение подобных оценок.

1. ХАРАКТЕРИСТИКА ОБЪЕКТА ДИАГНОСТИРОВАНИЯ

Структура избыточной МВС представляется графом G_0^n , вершинам которого соответствуют однородные группы ресурсов (процессоры, модули), а дугам — линии связи между ресурсами. Предполагается, что отказывать могут как вершины, так и дуги, причем отказ вершины приводит к отказу всех инцидентных ей (входящих и выходящих) дуг.

Алгоритмы синтеза [2] позволяют построить для n -вершинного целевого графа G_0^n и заданной степени m отказоустойчивости искомый объемлющий граф G_R^{n+m} . Наличие гамильтонова цикла в целевом графе G_0^n является необходимым условием существования отказоустойчивого графа G_R^{n+m} . Отметим некоторые свойства графа G_R^{n+m} , необходимые для последующих построений.



Граф G_R^{n+m} m -ОУ системы содержит $(n + m)$ вершин, из которых n вершин соответствуют рабочим модулям, а m вершин – резервные. Он симметричен относительно $(n + m)$ вращений (сдвигов) вокруг своего центра на угол $dg_1 = 360^\circ / (n + m)$, т. е. при повороте на угол dg_1 каждая вершина вместе с инцидентными ей дугами совмещается с соседней вершиной и инцидентными ей дугами. Такая симметрия обеспечивает замену отказавшей вершины и инцидентных ей дуг на исправную вершину и соответствующие дуги путем поворота на соответствующий угол, кратный dg_1 . Добавим, что граф G_R^{n+m} после подобной реконфигурации сохраняет логическую структуру задач, решавшихся на МВС до реконфигурации, так как старые логические имена компонентов, совпадавшие с их физическими номерами, будут заменены новыми, зависящими от выполненного числа сдвигов.

Для примера на рис. 1 показан исходный целевой граф G_0^8 , представленный в циркулянтном виде: внешние дуги образуют гамильтонов цикл между его восемью вершинами, а остальные дуги – внутренние. Двукратное применение алгоритмов синтеза [2] к графу G_0^8 привело к построению 2-ОУ графа G_R^{10} (рис. 2); подробности применения алгоритмов синтеза к графу G_0^8 можно найти в работе [5].

Для исправной m -ОУ системы, структура которой представлена графом G_R^{n+m} , существует C_{n+m}^m конфигураций (структур), включающих в себя n рабочих и m резервных процессоров (вершин графа). Для упрощения будем считать, что исходная рабочая конфигурация (РК) содержит n смежных рабочих вершин, номера которых совпадают с номерами целевого графа, а добавленные (резервные) вершины также являются смежными. Последовательно сдвигая $(n + m)$ вершин на одну позицию, например, против часовой стрелки, получим остальные $(n + m - 1)$ РК.

Для последующих действий представим информацию о графе G_R^{n+m} в виде таблицы, которую назовем первичной [5]. В ее левом столбце запишем номера $(n + m)$ РК, в следующих $(n + m)$ столбцах – составы этих РК, а затем – номера дуг, соответствующих номерам дуг исходных РК (т. е. номерам дуг целевого графа). Пример первичной таблицы, построенной для графа G_R^{10} , представлен табл. 1. Она содержит 8 столбцов для номеров рабочих вершин, 2 столбца – для номеров резервных вершин и 5 столбцов – для номеров дуг целевого графа (подробнее о составе этих дуг – при обсуждении примера).

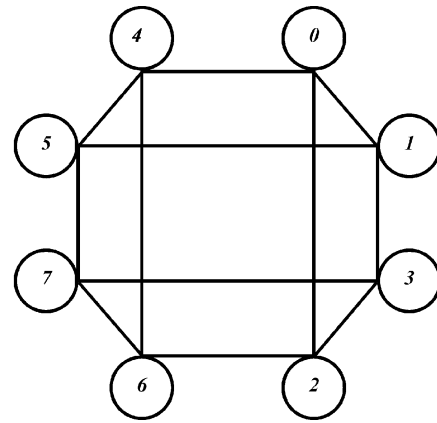


Рис. 1. Граф 3-куба

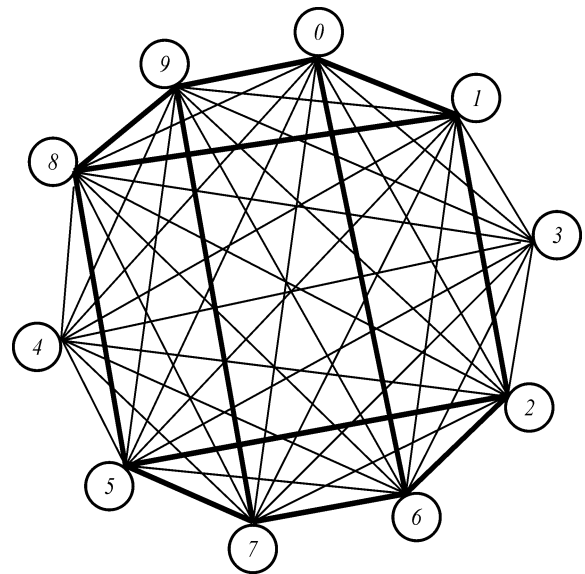


Рис. 2. Диагностический граф 2-ОУ системы

По мере функционирования m -ОУ системы и накопления в ней отказавших компонентов для их устранения будут использоваться РК с меньшим числом резервных вершин. Для систематического перебора таких РК нужна совокупность вторичных таблиц первого, второго, ..., m -го уровней, подобных первичной таблице, но содержащих меньше на одну, две, ..., m резервных вершин [5].

2. ПОСТАНОВКА ЗАДАЧИ

Пусть анализируемая m -ОУ система представлена диагностическим графом G_R^{n+m} , n вершин которого представляют n рабочих модулей (процессоров) системы, а m вершин – резервные и (или) отказавшие модули. По построению, граф G_R^{n+m} –

симметричный и однородный, но не обязательно полный.

Будем считать, что допускаются устойчивые кратные отказы ограниченного числа вершин и дуг. Предположим также, что во время выполнения СД новые отказы не возникают, а существующие неисправности таковы, что не приводят к несанкционированному запуску процесса СД в отказавших модулях и, следовательно, не искажают разработанную организацию СД; при реализации диагностических процедур используется модель БГМ [6].

Предполагается, что в исходной МВС есть диагностический монитор (часть центральной операционной системы), который инициирует процессы контроля, самодиагностирования и осуществляет реконфигурацию по результатам диагностирования технического состояния компонентов очередной РК.

Требуется разработать такой метод контроля и СД компонентов отказоустойчивой системы, чтобы после выполнения СД каждый исправный модуль имел информацию, достаточную для идентификации технического состояния остальных компонентов (модулей, линий связи) РК системы.

3. МЕТОД САМОДИАГНОСТИРОВАНИЯ ОТКАЗАВШИХ КОМПОНЕНТОВ

Контроль цифровой системы состоит в проверке каждым исправным модулем технического состояния своих преемников и соответствующих линий связи. Для оптимальной реализации последовательного выполнения различных проверок одного МПм разными МПщ на диагностическом графе системы построим гамильтонов путь, который, по определению, заходит в каждую вершину точно один раз. Если начальная и конечная вершины такого пути совпадают, то построенный путь

является гамильтоновым циклом. Для однородного диагностического графа исправной системы такой цикл всегда существует. Отметим, что для излагаемого ниже метода достаточно выбрать любой гамильтонов цикл, который обеспечивает обход всех вершин диагностического графа РК исправной системы. Такой цикл выбирается один раз при проектировании системы контроля.

Информацию о составе выбранной РК № w m -ОУ системы представим в виде таблицы T_{Sw} . Она состоит из n строк $T_{i_w}, \dots, T_j, \dots, T_{n_w}$, в каждой из которых для модуля m_j ($j = i_w, \dots, n_w$, где $0 \leq i_w, \dots, n_w = n + m - 1$) записаны номера модулей – преемников m_{k_1}, m_{k_2} , (смежных с модулем m_j), в ней также есть поля для записи результатов проверки модулем m_j этих модулей r_{j, k_1}, r_{j, k_2} , и значений их ранга при ранжировании графа. Пример такой таблицы T_{S0} , построенной для РК № 0 (из табл. 1), содержащей 8 строк, представлен в табл. 2. Как видно из табл. 2, в каждой ее строке заполнены только клетки для МПщ m_j и его преемников МПм – m_{k_1}, \dots, m_{k_3} .

Для отображения местонахождения модулей в гамильтоновом цикле будем записывать в строке T_j преемников модуля m_j так, что именно модуль m_{k_1} является соседом справа модуля m_j в построенном цикле. Для РК № 0 выбран такой гамильтонов цикл: $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 4 \rightarrow 0$, именно в таком порядке в табл. 2 записаны строки T_j .

Обычно процесс контроля запускается диагностическим монитором через определенные интервалы времени. Это предотвращает накопление отказавших компонентов в системе и способствует безошибочности решения целевых задач.

Назначим начальной вершиной процесса контроля одну из вершин выбранного гамильтонова

Таблица 1

Первичная таблица

№№ п/п	Номера рабочих вершин								Резервные вершины		Номера дуг				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	3	2	6	7	5	4*	8	9	4-0	0-2	3-7	6-4*	5-1*
1	1	3	2	6	7	5	4*	8	9	0	8-1	1-6	2-5	7-8	4-3
2	3	2	6	7	5	4*	8	9	0	1	9-3	3-7	6-4*	5-9	8-2
3	2	6	7	5	4*	8	9	0	1	3	0-2	2-5	7-8	4-0	9-6
4	6	7	5	4*	8	9	0	1	3	2	1-6	6-4*	5-9	8-1	0-7
5	7	5	4*	8	9	0	1	3	2	6	3-7	7-8	4-0	9-3	1-5*
6	5	4*	8	9	0	1	3	2	6	7	2-5	5-9	8-1	0-2	3-4
7	4*	8	9	0	1	3	2	6	7	5	6-4*	4-0	9-3	1-6	2-8
8	8	9	0	1	3**	2	6	7	5	4*	7-8	8-1	0-2	3-7	6-9**
9	9	0	1	3**	2	6	7	5	4*	8	5-9	9-3	1-6	2-5	7-0



цикла, обозначим ее m_n . Тогда по сигналу диагностического монитора МПЩ m_n должен выполнить следующие действия:

- из таблицы T_{Sw} выделить строку T_n , найти в ней номера преемников m_{k1}, m_{k2}, \dots ; проверить их техническое состояние и в клетки $r_{n, k1}, r_{n, k2}$ строки T_n записать результаты проверок в двоичном коде: 0 – исправен, 1 – подозревается в отказе;
- оценить результаты проверок:
 - если $r_{n, k1} = 0$, то передать управление процессом контроля модулю m_{k1} ;
 - если $r_{n, k1} = 1$, но один из других результатов (например, $r_{n, k2}$) – нулевой, то передать управление процессом контроля соответствующему модулю (модулю m_{k2});
 - если же проверки всех преемников m_n дали единичные результаты, то процесс контроля нужно начать с другого модуля (например, с модуля m_{n2} , находящегося в «середине» гамильтонова цикла).

Модуль $m_1 (m_{k1}, m_{k2}, \dots)$ и каждый из последующих модулей выполняет аналогичные действия. Этот процесс проверок и «хождения» по выбранному циклу заканчивается, когда управление возвращается к начальному модулю $m_n (m_{n2})$. Такой детерминированный путь обхода вершин графа возможен только при исправном состоянии всех компонентов проверяемой РК. Если же на одном из этапов проверки получен единичный результат, свидетельствующий об обнаружении одного или нескольких отказавших компонентов, то процесс контроля прерывается, и диагностический монитор переходит к процессу СД, который включает в себя такие процедуры:

- исполнение исправным модулем функций МПЩ;
- нахождение пути от одного исправного модуля к другому, возможно, с помощью процедуры

ранжирования соответствующих исправных модулей;

- обработка и дешифрация результатов выполненных проверок.

Эти процедуры реализуются совокупностью исправных модулей РК № w . Необходимость исполнения всех или части этих процедур определяется очередным МПЩ на основе результатов выполненных им проверок и информации, полученной от предыдущих модулей.

Для сохранения информации, полученной в процессе контроля, диагностический монитор начинает процесс СД с того исправного модуля $m_j (\neq m_n, m_{n2})$, который первым получил единичный результат. Начальный модуль m_j должен выполнить следующие действия:

- 1) образовать промежуточную таблицу PT_j и переписать в нее строку T_j с результатами проверок, полученными в процессе контроля (в индексе символа данной таблицы указан номер начального МПЩ);
- 2) по нулевым результатам проверок в строке T_j выделить подмножество номеров исправных преемников $\{q_j\}$ и одному из преемников передать таблицу PT_j и функции МПЩ на следующем шаге СД.

Если среди преемников m_j нет ни одного исправного, то процесс СД не может быть продолжен, и для получения диагностических результатов монитор должен запустить процесс контроля с другого модуля.

Очередной исправный модуль m_u , получив управление процессом СД, должен выполнить следующие действия:

- 1) из таблицы состава T_{Sw} выделить строку T_u , переписать ее в таблицу PT_j , полученную от предшественника $m_j (m_{u-1})$;

Таблица 2

Таблица состава T_{S0} для РК № 0

Символ строки	Проверяющий модуль и ранг		Преемники m_j , результаты их проверок и ранг								
	m_j	ранг m_j	m_{k1}	$r_{j, k1}$	ранг m_{k1}	m_{k2}	$r_{j, k2}$	ранг m_{k2}	m_{k3}	$r_{j, k3}$	ранг m_{k3}
T_0	0	–	1	–	–	2	–	–	4	–	–
T_1	1	–	3	–	–	5	–	–	0	–	–
T_3	3	–	2	–	–	7	–	–	1	–	–
T_2	2	–	6	–	–	0	–	–	3	–	–
T_6	6	–	7	–	–	4	–	–	2	–	–
T_7	7	–	5	–	–	3	–	–	6	–	–
T_5	5	–	4	–	–	1	–	–	7	–	–
T_4	4	–	0	–	–	6	–	–	5	–	–

2) в строке T_u найти номера своих преемников m_{k1}, m_{k2}, \dots ; проверить их состояние и в клетки $r_{u, k1}, r_{u, k2}$, строки T_u записать результаты проверок;

3) по нулевым результатам проверок в строке T_u выделить подмножество исправных преемников $\{q_u\}$, сравнить номера $\{q_u\}$ с номерами модулей в левом столбце PT_j :

- если один из номеров подмножества $\{q_u\}$ отсутствует в списке номеров левого столбца PT_j , то передать ему (модулю m_{u+1}) таблицу PT_j и функции МПЩ на следующем шаге; при этом модуль m_{u+1} вернется к п. 1 процесса СД;
- если два (или больше) номера из подмножества $\{q_u\}$ отсутствуют в списке номеров левого столбца PT_j , то одному из них (например, m_{k1}) передать таблицу PT_j и функции МПЩ на следующем шаге;
- если все номера $\{q_u\}$ в списке номеров левого столбца PT_j , то перейти к п. 4;

4) найти кратчайший путь от модуля m_u к одному из исправных модулей $m_{u+1}, m_{u+2}, \dots, m_{u+v}$, еще не исполнявших функции МПЩ, с помощью процедуры ранжирования, передать найденному модулю m_{u+1} (m_{u+2}, \dots) таблицу PT_j и функции МПЩ на следующем шаге; если при выполнении процедуры ранжирования выяснится, что все обнаруженные исправные модули исполняли функции МПЩ, перейти к п. 5;

5) передать информацию, собранную в таблице PT_j модуля m_{u+v} (у которого совпали списки исправных и проверяющих модулей), в направлении, обратном тому, по которому эта информация собиралась; получив идентичную информацию в виде таблицы PT_j от модуля m_{u+v} , каждый из модулей

выполнит процедуру ее обработки (приведенную ниже) и сформирует одинаковый диагноз.

Для нахождения кратчайшего пути от модуля m_u к одному из исправных модулей $m_{u+1}, m_{u+2}, \dots, m_{u+v}$, не являющихся его преемниками, каждому исправному модулю РК № w присваивается номер (ранг), равный минимальному числу линий связи, соединяющих данный модуль с модулем m_u , который ищет этот путь. Процедура ранжирования (присвоения номеров) носит «расширяющийся» характер: модуль m_u присваивает ранг 1 своим исправным преемникам, затем ранг 2 – исправным преемникам модулей с рангом 1, и т. д., пока некоторый ранг (например, d) не получит один из исправных модулей, еще не исполнявших функции МПЩ. Тогда поиск нужного пути заключается в выборе линий, связывающих модули с рангами d и $(d-1)$, затем модули с рангами $(d-1)$ и $(d-2)$ и т. д. до линии, связывающей модули с рангами 1 и 0 (m_u). Процедура формального описания действий модуля m_u при ранжировании графа и поиске нужного пути приведена в работе [14].

Для обработки и дешифрации информации, собранной в таблице PT_j модулем m_{u+v} , каждый исправный модуль переписывает ее в диагностическую таблицу T_{Dw} , построенную заранее для РК № w . В строке D_k ($k = i_w, \dots, n_w$) таблицы T_{Dw} для проверяемого модуля m_k записаны номера его предшественников (МПЩ) m_{j1}, m_{j2}, m_{j3} ; есть также клетки для записи результатов $r_{j1, k}, r_{j2, k}, r_{j3, k}$ его проверки этими модулями и клетки для записи диагностических заключений о состоянии модуля m_k и инцидентных ему линий связи, получаемых после обработки результатов $r_{j1, k}, r_{j2, k}, r_{j3, k}$. Пример такой таблицы, построенной для РК № 0 (из табл. 1), содержащей 8 строк, представлен в табл. 3. Нетрудно заметить, что преемники модуля m_j , по-

Таблица 3

 Диагностическая таблица T_{D0} для РК № 0

Символ строки	Предшественники m_k			Проверяемый модуль m_k	Результаты проверок			Диагностическое заключение			
	m_{j1}	m_{j2}	m_{j3}		$r_{j1, k}$	$r_{j2, k}$	$r_{j3, k}$	s_k	$a_{j1, k}$	$a_{j2, k}$	$a_{j3, k}$
D_0	1	2	4	0	0	0	–	0	0	0	–
D_1	3	5	0	1	0	1	0	0	0	1	0
D_3	2	7	1	3	0	0	0	0	0	0	0
D_2	6	0	3	2	0	0	0	0	0	0	0
D_6	7	4	2	6	0	–	0	0	0	–	0
D_7	5	3	6	7	0	0	0	0	0	0	0
D_5	4	1	7	5	–	1	0	0	–	1	0
D_4	0	6	5	4	1	1	1	1	z	z	z



казанные в строке T_j табл. 2, совпадают с предшественниками модуля m_k , приведенными в строке D_k табл. 3. Это совпадение – не случайность, а следствие использования в системе двунаправленных связей, поэтому модули m_{k1} , m_{k2} , m_{k3} – предшественники модуля m_j – являются также его предшественниками m_{j1} , m_{j2} , m_{j3} .

Процедура обработки исправным модулем информации из таблицы PT_j модуля m_{u+v} заключается в следующем:

1) из строк $T_j, \dots, T_u, \dots, T_{u+v}$ таблицы PT_j выделяются результаты проверок $r_{j, k1}, r_{j, k2}, \dots, r_{u, k1}, r_{u, k2}, \dots, r_{u+v, k1}, r_{u+v, k2}$ и записываются в клетки строк диагностической таблицы T_{Dw} ;

2) анализируются результаты проверок, содержащиеся в строке D_k , и в клетки $s_k, a_{j1, k}, a_{j2, k}, a_{j3, k}$ этой строки записывается диагностическое заключение, полученное согласно правилам:

- 1) если в клетках результатов проверок строки D_k есть хотя бы один нуль (пусть $r_{j1, k} = 0$), то в клетки на пересечении строки D_k со столбцами s_k и $a_{j1, k}$ записываются нули;
 - а) если в клетках результатов этой же строки D_k есть еще нули, то записываются нули в клетки на пересечении строки D_k со столбцами переменных соответствующих линий;
 - б) если в клетках результатов этой же строки D_k есть единицы, то записываются единицы в клетки на пересечении строки D_k со столбцами переменных соответствующих линий;
- 2) если в клетках результатов проверок строки D_k записаны только единицы, то записывается 1 в клетку на пересечении строки D_k со столбцом s_k , а в клетки на пересечении строки D_k со столбцами переменных соответствующих линий записываются буквы z .

Таким образом, дешифрация результатов проверок из таблицы PT_j модуля m_{u+v} сводится к замене 0, 1 и z , записанных в клетках $s_k, a_{j1, k}, a_{j2, k}, a_{j3, k}$ строки D_k таблицы T_{Dw} , состояниями модуля m_k и инцидентных линий связи согласно правилам:

- 0 → «модуль исправен», «линия исправна»;
- 1 → «модуль подозревается в отказе», «линия подозревается в отказе»;
- z → неопределенное состояние линии связи, так как отказавший модуль «маскирует» состояние инцидентной линии связи.

4. ПРОЦЕДУРА РЕКОНФИГУРАЦИИ

Дешифрация результатов проверок из таблицы PT_j модуля m_{u+v} определяет для РК № w подмножество компонентов, подозреваемых в отказе. Процедура реконфигурации m -ОУ графа сводится к нахождению в первичной таблице (или в одной из вторичных таблиц) такой строки, в которой компоненты, подозреваемые в отказе, не входят в состав РК № w^* – т.е. отказавшая вершина попадет на место одной из резервных, отказавшая дуга не будет соединять пару смежных рабочих вершин и не будет входить в состав дуг, перечисленных в правой части этой строки. Начало поиска нужной работоспособной РК зависит от того, к какой из таблиц – первичной или вторичной – принадлежит анализируемая РК № w .

После нахождения новой работоспособной РК № w^* диагностический монитор модифицирует таблицу состава T_{Sw} и диагностическую таблицу T_{Dw} , заменяя номера рабочих модулей прежней РК № w на соответствующие номера новой РК № w^* . Затем монитор запускает процесс контроля РК № w^* . Если контроль РК № w^* подтвердит ее работоспособность, то m -ОУ система возвращается к выполнению своих функциональных задач. В противном случае монитор вновь запускает процесс СД для идентификации тех компонентов, которые не участвовали в РК № w , а потому их отказы не влияли на ее работоспособность.

После идентификации нового подмножества подозреваемых компонентов процедура реконфигурации повторяется, но новая РК выбирается так, чтобы в ней отсутствовали не только впервые обнаруженные компоненты, подозреваемые в отказе, но и все отказавшие компоненты, ранее обнаруженные и выведенные из состава РК. Таким образом, может оказаться, что после отказа некоторого компонента в ПТ не останется ни одной работоспособной РК. В этом случае нужно перейти к РК, включающим в себя на 1, 2, ..., m вершин меньше. По определению, в каждую из таких РК с меньшим числом резервных вершин можно вложить целевой граф G_0^n . Но нужно проверить, что ранее отказавшие компоненты (вершины, дуги) не мешают работоспособности выбранной РК.

Несколько слов о стратегии перебора первичной и вторичных таблиц. Очевидно, что идентификация процессом СД отказавшей вершины вынуждает переходить к РК, не содержащим эту вершину; для этого может потребоваться переход к поиску во вторичных таблицах. Что же касается устранения влияния отказавшей дуги, то здесь целесообразно просмотреть все оставшиеся строки таблицы, РК которой функционировала в момент

обнаружения отказа. И если среди них нет ни одной работоспособной (вследствие влияния предыдущих отказов), то следует переходить к вторичной таблице с уменьшенным на единицу числом резервных вершин. При этом желательно исключить из дальнейшего рассмотрения ту вершину, которой инцидентно наибольшее число отказавших дуг.

Возможность возникновения неразличимых кратных отказов модулей и их входных линий связи [14] не является серьезным ограничением для восстановления работоспособности систем с реконфигурацией. Действительно, смена РК выводит из числа рабочих не только подозреваемый модуль, но и все его входные и выходные линии связи. Поэтому для таких систем диагностирование отказавших компонентов с точностью до модуля или его входной линии связи представляется необходимым.

5. ПРИМЕР

Особенности процесса СД и реконфигурации покажем на примере 2-ОУ системы, построенной для целевой системы со структурой в виде трехмерного куба. Гиперкубовые структуры МВС обладают рядом достоинств, среди которых отмечают простоту алгоритмов маршрутизации при межпроцессорных обменах информацией, регулярность алгоритмов реконфигурации систем при отказах их компонентов, хорошо формализованные модели их представления [15, 16]. В примере рассматривается 3-куб, в вершинах которого расположены процессорные элементы (на рис. 1 они обозначены кружками, в которых указаны их номера).

Считая граф 3-куба (см. рис. 1) в качестве целевого графа G_0^8 , путем двукратного применения алгоритмов синтеза [2] был построен 2-отказоустойчивый граф G_R^{10} (см. рис. 2). Первичная таблица РК графа G_R^{10} (см. табл. 1) состоит из 10-ти строк, соответствующих 10-ти РК и обозначенных номерами 0, 1, ..., 9, клетки столбцов 1, 2, ..., 8 в каждой строке содержат номера рабочих вершин данной РК, клетки столбцов 9 и 10 – номера резервных вершин. В столбцах 12–15 приведены номера дуг, соответствующие номерам четырех дуг целевого графа G_0^8 . Начальная РК № 0 показана в строке 0: рабочие 0, 1, ..., 5, 4 и резервные 8, 9 вершины, рабочие дуги (0–2), (3–7), (6–4) и (5–1). В столбце 11 показана дуга (4–0): она замыкает гамильтонов цикл, в который входят рабочие вершины 0, 1, ..., 5, 4. Дуги между смежными вершинами этого цикла в табл. 1 отдельно не показаны, так как их номера однозначно определя-

ются номерами соседних столбцов: (0–1), (1–3) и т. д. Номера вершин РК, показанные в строках 1–9, получены циклическим сдвигом справа налево номеров вершин строки 0. Номера дуг, показанные в клетках столбцов 11–15 в каждой строке, получены путем замены номеров рабочих вершин строки 0 номерами вершин, записанными в тех же столбцах очередной РК. Поэтому в столбце 12 дуга (0–2) из строки 0 последовательно заменяется дугой, соединяющей вершины с номерами (1–6), (3–7) и т. д., стоящими в соответствующих столбцах ниже вершин 0 и 2. Таким же образом получены номера других рабочих дуг.

В статье [15] предложен алгоритм А2 оптимального обхода и тестирования вершинных процессорных элементов, состоящий из $2n$ раундов параллельных проверок смежных вершин; n -размерность куба. Таким образом, все 24 теста, необходимые для контроля состояния 3-куба, будут выполнены за 6 раундов, т.е. в 4 раза быстрее последовательного выполнения этих тестов. Но в реализации этих тестов участвуют все модули n -куба, как исправные, так и отказавшие. Поэтому таблица проверок n -куба с отказавшими компонентами будет содержать недостоверные результаты, что усложнит процедуру дешифрации.

Подчеркнем, что алгоритм А2 разработан для проверки работоспособности вершин и дуг неизбыточного n -куба. Поэтому его применение для проверки 2-ОУ системы, построенной на основе 3-куба, возможно только для отдельных РК, содержащих 8 вершин и 12 дуг. Нетрудно видеть, что 2-ОУ граф G_R^{10} (см. рис. 2) содержит большое число таких РК. Поэтому вместо перебора всевозможных РК будем использовать описанный процесс последовательного контроля компонентов РК № 0 и тех РК, в которые переходит система после обнаружения и устранения очередного подмножества отказавших компонентов.

Итак, пусть 2-ОУ система работает в начальной РК № 0 (см. табл. 1). Предположим, что из-за большого интервала между циклами контроля в 2-ОУ системе накопились отказы модуля 4 и линий связи (5–1) и (6–4) (эти компоненты отмечены знаком * в клетках тех РК табл. 1, в состав которых они входят). Контроль работоспособности 2-ОУ системы начинается с модуля, соответствующего начальной вершине гамильтонова цикла для выбранной РК. Так, для РК № 0 выберем такой цикл: 0 → 1 → 3 → 2 → 6 → 7 → 5 → 4 → 0.

По сигналу диагностического монитора начальный модуль процесса контроля m_0 (вершина 0) выполнит следующие действия:

– из таблицы T_{S_0} состава РК № 0 (см. табл. 2) выделит строку T_0 , найдет в ней номера приемни-



ков m_{k1} (1), m_{k2} (2), m_{k3} (4), проверит их техническое состояние и в клетки $r_{j, k1}$, $r_{j, k2}$, $r_{j, k3}$ строки T_0 запишет результаты проверок: $r_{0,1} = 0$, $r_{0,2} = 0$, $r_{0,4} = 1$ которые показаны в строке T_0 табл. 4;

– оценит результаты проверок: хотя $r_{0,4} = 1$, но модуль 0 начальный, поэтому модуль 0 передаст управление процессом контроля соседнему модулю 1.

Модуль 1 (m_{k1}), получив управление от модуля 0, выполнит следующие действия:

– из таблицы T_{S0} выделит строку T_1 , найдет в ней номера преемников 3, 5, 0, проверит их и в соответствующие клетки строки T_1 запишет результаты проверок: $r_{1,3} = 0$, $r_{1,5} = 1$, $r_{1,0} = 0$ (для сокращения числа таблиц будем записывать результаты

проверок, находящиеся в разных модулях, в общую табл. 4);

– оценив результаты проверок и обнаружив $r_{1,5} = 1$, достоверно исправный модуль 1 перейдет к процессу СД.

Для сохранения информации, полученной в процессе контроля, монитор начнет процесс СД с модуля 1, который выполнит следующие действия:

1) образует промежуточную таблицу PT_1 (см. табл. 4), в которую переписит строку T_1 с результатами проверок, полученными в процессе контроля (в индексе символа данной таблицы указан номер 1 начального МПц);

2) по нулевым результатам проверок в строке T_1 выделит номера 3 и 0 исправных преемников $\{q_1\}$ и согласно следующему пункту процедуры пере-

Таблица 4

Информация к Примеру

Символ строки	Проверяющий модуль		Преемники m_j , результаты их проверок и ранг								
	m_j	ранг m_j	m_{k1}	$r_{j, k1}$	ранг m_{k1}	m_{k2}	$r_{j, k2}$	ранг m_{k2}	m_{k3}	$r_{j, k3}$	ранг m_{k3}
Строки с результатами контроля РК № 0											
T_0	0	–	1	0	–	2	0	–	4	1	–
T_1	1	–	3	0	–	5	1	–	0	0	–
Таблица PT_1											
T_1	1	–	3	0	–	5	1	*	0	0	–
T_0	0	–	1	0	–	2	0	–	4	1	–
T_2	2	–	6	0	–	0	0	–	3	0	–
T_6	6	2	7	0	1	4	1	*	2	0	3
T_7	7	1	5	0	0	3	0	2	6	0	2
T_5	5	0	4	1	*	1	1	*	7	0	1
T_3	3	2	2	0	–	7	0	–	1	0	–
Строки с результатами контроля РК № 8											
T_8	8	–	9	0	–	1	0	–	7	0	–
T_9	9	–	0	0	–	6	1	–	8	0	–
Таблица PT_9											
T_9	9	–	0	0	–	6	1	–	8	0	–
T_0	0	–	1	0	–	2	0	–	9	0	–
T_1	1	–	3	1	–	8	0	–	0	0	–
T_8	8	–	9	0	–	1	0	–	7	0	–
T_7	7	–	8	0	–	3	1	–	6	0	–
T_6	6	–	7	0	–	9	1	–	2	0	–
T_2	2	–	6	0	–	0	0	–	3	1	–

даст таблицу PT_1 и функции МПЩ на следующем шаге СД, например, модулю θ .

Модуль θ выполнит следующие действия:

1) из таблицы T_{S0} выделит строку T_0 и переписет ее в таблицу PT_1 (см. табл. 4);

2) в строке T_0 найдет номера преемников $1, 2, 4$; проверит их и в соответствующие клетки строки T_0 запишет результаты проверок: $r_{0,1} = 0, r_{0,2} = 0, r_{0,4} = 1$;

3) по нулевым результатам проверок в строке T_0 выделит номера исправных преемников $\{q_0 = 1, 2\}$, сравнит их с номерами $1, \theta$ в левом столбце таблицы PT_1 и обнаружит, что модуль 2 не исполнял функции МПЩ, поэтому передаст ему таблицу PT_1 с результатами проверок и функции МПЩ на следующем шаге СД, при этом модуль 2 вернется к п. 1 (см. § 3) процесса СД.

Аналогичные действия в роли МПЩ поочередно выполняют модули 2, 6, 7 и 5, результаты их проверок своих преемников показаны в строках T_2, T_6, T_7, T_5 табл. 4. В отличие от предыдущих МПЩ модуль 5 обнаружит, что его исправный преемник – модуль 7 – уже исполнял функции МПЩ и перейдет к поиску пути к одному из исправных модулей, еще не исполнявших функции МПЩ, с помощью процедуры ранжирования.

Из результатов проверок, записанных в строках T_1, \dots, T_5 табл. PT_1 , модуль 5 определит, что только исправный модуль 3 не исполнял функции МПЩ, из табл. T_{S0} выделит строку T_3 и переписет ее в табл. PT_1 . Затем, выполняя процедуру ранжирования, модуль 5 присвоит ранг θ модулю 5, ранг 1 модулю 7, затем ранг 2 модулям 3 и 6. Ранжирование закончено, так как модуль 3 попал в число ранжированных. (Результаты ранжирования показаны в строках табл. PT_1 ; в некоторых клетках столбцов «ранг m_{k1} » и «ранг m_{k2} » стоит знак *, означающий, что соответствующие модули m_{k1} и m_{k2} не участвуют в формировании искомого пути, так как они признаны отказавшими по единичным результатам их проверок). Теперь модуль 5 передаст табл. PT_1 и функции МПЩ на следующем шаге СД модулю 3 через цепочку модулей: $5 \rightarrow 7 \rightarrow 3$.

Модуль 3, получив управление от модуля 5, выполнит следующие действия:

1) в строке T_3 табл. PT_1 найдет номера преемников $2, 7, 1$; проверит их и в соответствующие клетки строки T_3 запишет результаты: $r_{3,2} = r_{3,7} = r_{3,1} = 0$;

2) по нулевым результатам проверок в строке T_3 выделит номера исправных преемников $\{q_3 = 2, 7, 1\}$,

обнаружит, что все они входят в список номеров левого столбца PT_1 , что других исправных модулей, не исполнявших функции МПЩ, нет, поэтому процесс сбора тестовой информации закончен. Теперь модуль 3 отправит табл. PT_1 в обратный путь по цепочке: $3 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow 2 \rightarrow \theta \rightarrow 1$, чтобы каждый из исправных модулей имел одинаковую информацию.

Получив табл. PT_1 от модуля 3, каждый исправный модуль выделит из ее строк результаты проверок и запишет их в клетки строк диагностической таблицы T_{D0} (см. табл. 3). Затем на основании правил, приведенных в основном тексте статьи, сформирует и запишет в клетки $s_k, a_{j1,k}, a_{j2,k}, a_{j3,k}$ восьми строк табл. T_{D0} соответствующие диагностические заключения. Их дешифрация позволяет сделать следующие выводы:

– подозреваются в отказе модуль 4 и линии связи $(5 - 1)$ и $(1 - 5)$;

– нет информации о состоянии линий $(\theta - 4)$, $(6 - 4)$ и $(5 - 4)$, так как отказавший модуль 4 «маскирует» отказы своих входных линий, поэтому линия $(6 - 4)$ не попала в число подозреваемых в отказе;

– также нет информации о состоянии линий $(4 - \theta)$, $(4 - 6)$ и $(4 - 5)$: отказавший модуль 4 не использовался в качестве МПЩ;

– в РК № 0 исправны 7 модулей $(\theta, 1, \dots, 7, 5)$ и остальные линии связи, кроме перечисленных.

Сравнение полученного диагноза с составом исходной отказовой ситуации показывает, что вместо линии $(5 - 1)$ подозревается в отказе пара линий $(5 - 1)$ и $(1 - 5)$; причина такого диагноза в том, что логическими методами нельзя различить отказы линий, участвующих в инверсных проверках.

Для устранения влияния подозреваемых компонентов – модуля 4 и дуг $(5 - 1)$ и $(1 - 5)$ – монитор должен перевести 2-ОУ систему из РК № 0 в РК, в которую не входят эти компоненты. Обратившись к первичной таблице (см. табл. 1), нетрудно обнаружить, что модуль 4 входит в состав восьми РК с номерами от 0 до 7 и не входит только в РК № 8 и 9, где он записан в числе резервных модулей; в эти РК не входит и дуга $(5 - 1)$. Для определенности, выберем в качестве следующей РК № 8.

После нахождения новой работоспособной РК диагностический монитор должен модифицировать таблицу состава T_{S0} и диагностическую таблицу T_{D0} , заменив номера рабочих модулей в соответствии с подстановкой: (89013267) вместо (01326754) и получив таблицу T_{S8} (табл. 5) и диагностическую таблицу T_{D8} (табл. 6). Затем монитор



запустит процесс контроля РК № 8, который подтвердит ее работоспособность.

Пусть спустя некоторое время работы системы в РК № 8 возникнут отказы модуля 3 и линии (6 – 9), которые в табл. 1 отмечены знаком **.

Как отмечалось, контроль работоспособности 2-ОУ системы начинается с модуля, соответствующего начальной вершине гамильтонова цикла для выбранной РК. После смены РК и замены номеров рабочих вершин меняется и состав гамильтонова цикла: так, для РК № 8 получим такой цикл: $8 \rightarrow 9 \rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 8$. Теперь по сигналу ДМ начальный модуль 8 процесса контроля выполнит действия, аналогичные действиям модуля 0 в РК № 0, но искомую строку T_8 возьмет из табл. T_{S8} , передаст управление контролем модулю 9, который, обнаружив единичный результат проверки модуля 6, переведет ДМ в режим выполнения процесса СД (результаты проверок, выпол-

ненных модулями 8 и 9, показаны в строках T_8 и T_9 табл. 4).

Начальный модуль процесса СД – модуль 9 – начнет построение промежуточной таблицы PT_9 , переписав в нее строку T_9 с результатами проверок, выполненных в процессе контроля, затем по нулевым результатам проверок выделит номера 0 и 8 исправных преемников, и одному из них, например, модулю 0 передаст таблицу PT_9 и функции МПЩ на следующем шаге СД.

Действия модуля 0 и последующих МПЩ аналогичны действиям, ранее описанным в процессе СД в РК № 0, за исключением того, что необходимые строки будут извлекаться из табл. T_{S8} (см. табл. 5); результаты выполненных ими проверок показаны в табл. PT_9 (см. табл. 4). Модуль 2, завершив ее построение, обнаружит, что больше нет исправных модулей, не выполнявших функции МПЩ, и передаст табл. PT_9 по обратной цепочке:

Таблица 5

Таблица состава T_{S8} для РК № 8

Символ строки	Проверяющий модуль		Преемники m_j , результаты их проверок и ранг								
	m_j	ранг m_j	m_{k1}	$r_{j, k1}$	ранг m_{k1}	m_{k2}	$r_{j, k2}$	ранг m_{k2}	m_{k3}	$r_{j, k3}$	ранг m_{k3}
T_8	8	–	9	–	–	1	–	–	7	–	–
T_9	9	–	0	–	–	6	–	–	8	–	–
T_0	0	–	1	–	–	2	–	–	9	–	–
T_1	1	–	3	–	–	8	–	–	0	–	–
T_3	3	–	2	–	–	7	–	–	1	–	–
T_2	2	–	6	–	–	0	–	–	3	–	–
T_6	6	–	7	–	–	9	–	–	2	–	–
T_7	7	–	8	–	–	3	–	–	6	–	–

Таблица 6

Диагностическая таблица T_{D8} для РК № 8

Символ строки	Предшественники m_k			Проверяемый модуль m_k	Результаты проверок			Диагностическое заключение			
	m_{j1}	m_{j2}	m_{j3}		$r_{j1, k}$	$r_{j2, k}$	$r_{j3, k}$	s_k	$a_{j1, k}$	$a_{j2, k}$	$a_{j3, k}$
D_8	9	1	7	8	0	0	0	0	0	0	0
D_9	0	6	8	9	0	1	0	0	0	1	0
D_0	1	2	9	0	0	0	0	0	0	0	0
D_1	3	8	0	1	–	0	0	0	–	0	0
D_3	2	7	1	3	1	1	1	1	z	z	z
D_2	6	0	3	2	0	0	–	0	0	0	–
D_6	7	9	2	6	0	1	0	0	0	1	0
D_7	8	3	6	7	0	–	0	0	0	–	0

$2 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 1 \rightarrow 0 \rightarrow 9$. Каждый из этих модулей переписывает из PT_9 результаты проверок в диагностическую таблицу T_{D8} (см. табл. 6), обрабатывает их и придет к следующим диагностическим заключениям:

- подозреваются в отказе модуль 3 и линии связи $(6 - 9)$ и $(9 - 6)$;
- нет информации о состоянии линий $(2 - 3)$, $(7 - 3)$ и $(1 - 3)$, так как отказавший модуль 3 «маскирует» отказы своих входных линий;
- также нет информации о состоянии линий $(3 - 1)$, $(3 - 2)$ и $(3 - 7)$: отказавший модуль 3 не использовался в качестве МПЩ;
- в РК № 8 исправны семь модулей $(9, 0, 1, 3, 2, 6, 7)$ и остальные линии связи, кроме перечисленных.

Сравнение полученного диагноза с составом исходной отказовой ситуации показывает, что вместо линии $(6 - 9)$ подозревается в отказе пара линий $(6 - 9)$ и $(9 - 6)$; причина – та же, что и при диагностировании линии $(5 - 1)$ в РК № 0.

Для устранения влияния подозреваемых компонентов – модуля 3 и дуг $(6 - 9)$ и $(9 - 6)$ – монитор должен перевести 2-ОУ систему из РК № 8 в РК, в которую не входят эти компоненты, а также ранее обнаруженные отказавшие модуль 4 и линии $(5 - 1)$ и $(1 - 5)$. Отказ двух модулей оставляет в составе 2-ОУ системы восемь исправных модулей, т. е. система оказывается избыточной для отказов любого рабочего модуля. Удаление из графа G_R^{10} (см. рис. 2) отказавших вершин 4 и 3 и инцидентных им дуг преобразует его в восьми вершинный полно связный граф $G_{R^*}^8$, показанный на рис. 2 более толстыми линиями. В статьях [4, 5] показано, что избыточное число дуг длины 3 в графе $G_{R^*}^8$ по сравнению с числом дуг длины 3 в целевом графе G_0^8 предоставляет дополнительные возможности для устранения их отказов с помощью реконфигурации. Действительно, в графе G_0^8 таких дуг – четыре, их номера в РК № 0 – $(0 - 2)$, $(3 - 7)$, $(6 - 4)$ и $(5 - 1)$; а граф $G_{R^*}^8$ содержит восемь дуг длины 3. Анализ расположения этих восьми дуг показывает, что они позволяют построить два РК из восьми вершин с непересекающимися подмножествами дуг длины 3. На рис. 2 показана одна из этих РК, использующая дуги длины 3 с номерами $(2 - 5)$, $(7 - 9)$, $(8 - 1)$ и $(0 - 6)$. Вторая РК использует дуги длины 3 с номерами $(6 - 8)$, $(5 - 0)$, $(9 - 2)$, $(1 - 7)$. Нетрудно проверить, что одно подмножество дуг переходит в другое (и наоборот) при сдвиге рабочих вершин графа $G_{R^*}^8$ на 1 шаг, 3, 5, 7 шагов (2, 4 шага, 6, 8 шагов) по га-

мельтонову циклу для графа $G_{R^*}^8$: $2 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 0 \rightarrow 1 \rightarrow 2$. Следовательно, наличие двух работоспособных РК в графе $G_{R^*}^8$ позволяет устранить отказ одной дуги, входящей (или отказы 2...4 дуг, входящих) в одно из приведенных подмножеств, путем реконфигурации и таким образом продлить рабочее функционирование 2-ОУ системы без ручной замены отказавших линий связи.

ЗАКЛЮЧЕНИЕ

Предложен метод самодиагностирования состояния компонентов (модулей и линий связи) цифровых систем с реконфигурацией, для построения графов структур которых применены алгоритмы синтеза, предложенные М.Ф. Караваем. В основу разработанного метода СД положен принцип расширяющихся областей: выполнение необходимых процедур начинается с исправного модуля и последовательно передается другим модулям, исправность которых подтверждена предшествующими проверками. Начальный модуль процедуры СД не является фиксированным, а определяется по результатам работы системы контроля. Предполагается, что допускаются устойчивые кратные отказы ограниченного числа компонентов, причем за время диагностирования новые отказы не возникают, а существующие неисправности таковы, что не приводят к несанкционированному запуску процесса СД в отказавших модулях. Диагностические процедуры реализуются в соответствии с моделью БГМ.

Предполагается, что в исходной отказоустойчивой системе есть диагностический монитор (часть центральной операционной системы), который инициирует процессы контроля, СД и осуществляет реконфигурацию по результатам диагностирования очередной РК.

В соответствии с рассмотренным методом СД выполняются процедуры: исполнения исправным модулем функций проверяющего для своих приемников; нахождения кратчайшего пути от исправного модуля к ближайшему, исправному модулю, еще не исполнявшему функций проверяющего; обработки и дешифрации исправным модулем результатов выполненных проверок.

Для устранения влияния компонентов, подозреваемых в отказе, путем смены РК отказоустойчивого графа применяется способ реконфигурации, представляющий собой вариант скользящего резервирования. В цифровых системах возможны такие сочетания отказавших смежных компонентов (модулей и линий связи), потенциальные синдромы которых неразличимы для однократного



диагностирования. Однако для отказоустойчивых систем с реконфигурацией неразличимость отказов смежных компонентов не препятствует восстановлению их работоспособности.

ЛИТЕРАТУРА

1. *Авиженис А.* Отказоустойчивость — свойство, обеспечивающее постоянную работоспособность цифровых систем // ТИИЭР. — 1978. — Т. 66. — № 10. — С. 5–25.
2. *Каравай М.Ф.* Инвариантно-групповой подход к исследованию k -отказоустойчивых структур // Автоматика и телемеханика. — 2000. — № 1. — С. 144–156.
3. *Maehle E., Moritzen K., Wirl K.* A graph model for diagnosis and reconfiguration and its application to a fault-tolerant multi-processor system // 16-th Int. Conf. on Fault-Tolerant Computing FTCS-16, 1986. — P. 292–297.
4. *Ведешенков В.А.* Об использовании избыточных хорд k -отказоустойчивого графа для устранения влияния отказавших компонент // Автоматика и телемеханика. — 2003. — № 4. — С. 158–166.
5. *Ведешенков В.А.* Процедура восстановления работоспособности отказоустойчивых цифровых систем с динамической избыточностью // Автоматика и телемеханика. — 2003. — № 5. — С. 167–179.
6. *Barsi F., Grandoni F., Maestrini P.* A theory of diagnosability of digital systems // IEEE Trans. Comput. — 1976. — Vol. C-25. — № 6. — P. 585–593.
7. *Preparata F.P., Metzger G., Chien R.T.* On the connection assignment problem of diagnosable systems // IEEE Trans. Electr. Comput. — 1967. — Vol. 16. — № 6. — P. 848–854.
8. *Пархоменко П.П.* Определение технического состояния многопроцессорных вычислительных систем путем анализа графа синдромов // Автоматика и телемеханика. — 1999. — № 5. — С. 126–135.
9. *Buskens R.W., Bianchini R.P.* Distributed on-line diagnosis in the presence of arbitrary faults // Proc. of the 23-rd Int. Symp. on Fault Tolerant Computing, 1993. — P. 470–479.
10. *Hosseini S.H., Kuhl J.G., Reddy S.M.* A diagnosis algorithm for distributed computing systems with dynamic failure and repair // IEEE Trans. Comput. — 1984. — Vol. C-33. — № 3. — P. 223–233.
11. *Bianchini R.Jr., Goodwin K., Nydick D.S.* Practical application and implementation of distributed system-level diagnosis theory // Proc. of the 20-th Int. Symp. on Fault Tolerant Computing, 1990. — P. 332–339.
12. *Bianchini R.Jr., Buskens R.W.* An adaptive distributed system-level diagnosis algorithm and its implementation // Proc. of the 21-st Int. Symp. on Fault Tolerant Computing, 1991. — P. 222–229.
13. *Ведешенков В.А.* Организация самодиагностирования технического состояния цифровых систем // Автоматика и телемеханика. — 2003. — № 11. — С. 165–182.
14. *Ведешенков В.А.* О диагностировании отказавших модулей и связей в цифровых системах с использованием модели БГМ // Автоматика и телемеханика. — 2002. — № 2. — С. 159–171.
15. *Пархоменко П.П.* О системном диагностировании вершинных и реберных гиперкубовых структур многопроцессорных вычислительных систем // Автоматика и телемеханика. — 1996. — № 11. — С. 165–174.
16. *Ведешенков В.А.* О системном диагностировании отказавших модулей и связей в гиперкубовых структурах многопроцессорных вычислительных систем // Автоматика и телемеханика. — 2001. — № 12. — С. 83–94.

☎ (095) 334-75-90

E-mail: vva@i9.ipu.rssi.ru



Новая книга

Новиков Д.А. Стимулирование в организационных системах. — М.: СИНТЕГ, 2003. — 312 с. (серия «Управление организационными системами»).

Монография посвящена описанию формальных моделей стимулирования в организационных системах. Ее целью, помимо ознакомления читателя с современным состоянием дел в данной области теории управления, является демонстрация необходимости и целесообразности применения моделей стимулирования для повышения эффективности работы организаций.

Рассмотрены модели индивидуального и коллективного стимулирования, установлена взаимосвязь между результатами моделирования и применяемыми на практике формами и системами оплаты труда, приведены результаты экспериментальных исследований, обсуждена возможность использования решений задач стимулирования при оптимизации состава организационных систем.

Книга адресована студентам вузов, аспирантам и специалистам (теоретикам и практикам) в области экономики труда и управления социально-экономическими системами.

Д.А. Новиков — д-р техн. наук, профессор кафедры «Проблемы управления» Московского физико-технического института, ведущий научный сотрудник лаборатории активных систем Института проблем управления им. В.А. Трапезникова. Автор многих работ по теории управления социально-экономическими системами, в том числе по теории игр, принятию решений и механизмам управления организационными системами.