

САМОДИАГНОСТИРОВАНИЕ ВОЗНИКАЮЩИХ ОТКАЗОВ И ИХ УСТРАНЕНИЕ В ЦИФРОВЫХ СИСТЕМАХ С РЕКОНФИГУРАЦИЕЙ

В. А. Ведешенков

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

Предложен метод самодиагностирования (СД) и устранения возникающих отказов компонентов (модулей и линий связи) цифровых систем с реконfigurацией. Допускаются устойчивые кратные отказы ограниченного числа компонентов, причем в процессе диагностирования может возникнуть одиночный отказ. Для обнаружения таких отказов и выработки правильного диагноза осуществляется дву- и более кратный запуск процесса СД, результаты двух последних запусков сравниваются. Для устранения отказавших компонентов путем смены рабочих конфигураций применяется вариант скользящего резервирования. Дан пример реализации СД одиночного отказа, возникшего в процессе СД, и реконfigurации 2-отказоустойчивой системы.

ВВЕДЕНИЕ

Многопроцессорные вычислительные системы (МВС) широко применяются в автоматизированных системах управления различными объектами. Необходимая отказоустойчивость МВС часто достигается с помощью стратегии динамической избыточности, включающей в себя этапы контроля работоспособности системы, идентификации отказавшего компонента, реконfigurации для его замены исправным компонентом из числа резервных, восстановления функционирования системы [1, 2]. Методы синтеза графа структуры отказоустойчивых систем, предложенные в работе [2], позволяют построить граф с заданной степенью m отказоустойчивости (m -ОУ граф) для целевого (исходного избыточного) графа достаточно общего вида. Применение инвариантно-группового подхода приводит к получению структур m -ОУ графов регулярного вида, что упрощает процедуры их диагностирования и реконfigurации. Но работ по диагностированию отказавших компонентов и реконfigurации в отказоустойчивых системах немного [3–6], причем в работе [6] основное внимание было уделено разработке метода СД устойчивых отказов компонентов цифровых систем с реконfigurацией.

В настоящей статье при тех же предпосылках об объекте исследуются особенности СД тех компо-

нентов, которые отказывают во время процесса СД. Результаты взаимного тестирования отдельных частей системы (модулей), как и в работе [6], описываются с помощью модели Барси—Грандони—Маестрини (сокращенно, модели БГМ) [7]. Суммарный результат проверки модулей системы образует двоичный вектор R , который называется реальным синдромом [7, 8]. Поскольку в данной работе тестовая информация будет представляться в табличном виде, то в дальнейшем будем говорить об обработке табличной информации.

Ниже представлен метод СД возникающих отказов цифровых систем с реконfigurацией, опирающийся на принцип расширяющихся областей, который состоит в том, что выполнение необходимых процедур начинается с исправного модуля и последовательно передается другим модулям, исправность которых подтверждена предшествующими проверками. Основой применения этого принципа служит следующая гипотеза модели БГМ: нулевой результат, выдаваемый проверяющим модулем (МПШ), подтверждает исправность проверенного им модуля, поэтому для идентификации исправных модулей достаточно 0-путей единичной длины (в терминологии работы [9]). Добавим, что начальный модуль СД не является фиксированным, а определяется по результатам работы системы контроля [10].



СВЕДЕНИЯ ОБ ОБЪЕКТЕ 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 исходящих) 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} .

Граф G_R^{n+m} m -отказоустойчивой (ОУ) системы содержит $(n + m)$ вершин, из которых n вершин соответствуют рабочим модулям, а оставшиеся m вершин — резервные. Он симметричен относительно $(n + m)$ поворотов (сдвигов) вокруг центра графа 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 графа G_R^{n+m} обеспечивает замену отказавшей вершины и инцидентных ей дуг на исправную вершину и инцидентные ей дуги путем поворота на соответствующий угол, кратный dg_1 .

Обозначим символом $c_v(c_a)$ максимальное число отказавших вершин (внешних дуг) m -ОУ графа G_R^{n+m} , которые могут быть исключены из состава работающей системы путем смены рабочей конфигурации (РК); тогда справедливо соотношение

$$c_v + c_a \leq m. \tag{1}$$

Соотношение $c_v \leq m$ при $c_a = 0$ очевидно, так как, по определению, граф G_R^{n+m} является m -ОУ относительно отказов m его вершин. Для устранения отказавшей внешней дуги, соединяющей соседние вершины гамильтонова цикла, необходимо вывести из состава РК (и не использовать в последующих РК) хотя бы одну из этих вершин, т. е., по существу, считать ее отказавшей. Таким образом, для устранения c_a отказавших внешних дуг нужно исключить c_a вершин, если эти дуги не имеют общих вершин. Неравенство в соотношении (1) учитывает влияние возможных отказов внутренних дуг, которые могут понизить оценки для чисел c_v и c_a .

Для примера на рис. 1 показан исходный целевой граф G_0^8 : внешние дуги образуют гамильтонов

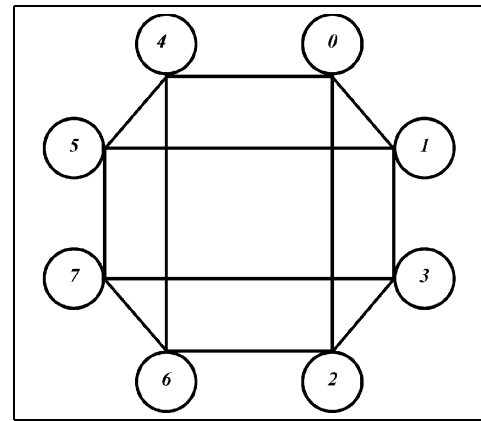


Рис. 1. Граф 3-куба

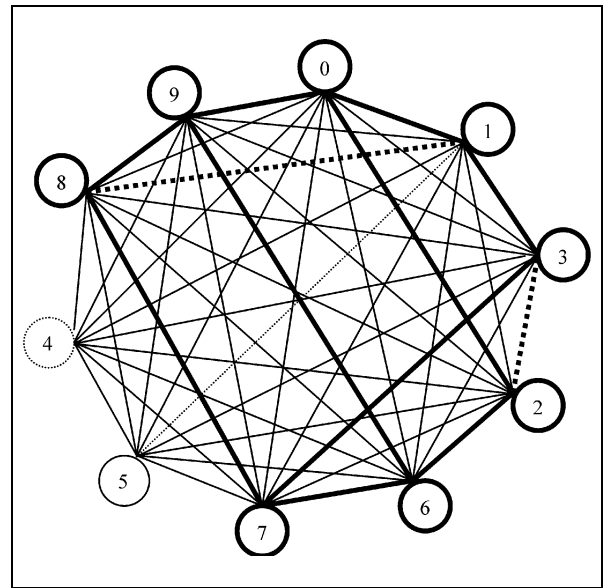


Рис. 2. Диагностический граф рабочей конфигурации № 8 2-ОУ системы

цикл между его восемью вершинами, а остальные дуги — внутренние. Двукратное применение алгоритмов синтеза [2] к графу G_0^8 позволяет построить 2-ОУ граф G_R^{10} (рис. 2) (смысл линий различной толщины, изображающих вершины и дуги, будет пояснен в Примере).

Каждая из конфигураций (структур) исправной m -ОУ системы, представленной графом G_R^{n+m} , включает в себя n рабочих и m резервных модулей (вершин графа). Для упрощения будем считать, что исходная 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, напри-

мер, 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, соответствующие номерам дуг исходной РК (т. е. номерам дуг целевого графа). Пример первичной таблицы, построенной для графа G_R^{10} , представлен в табл. 1. Она содержит восемь столбцов для номеров рабочих вершин, два столбца — для резервных вершин и пять столбцов — для номеров дуг целевого графа (подробнее о составе этих дуг — при обсуждении примера).

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

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 оставшихся 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. В отличие от работы [6] предполагается, что во время выполнения СД может возникнуть одиноч-

ный отказ. Предполагается, что неисправности модулей таковы, что не приводят к несанкционированному запуску СД в отказавших модулях и, следовательно, не искажают разработанную организацию СД.

Предполагается, что в исходной МВС есть диагностический монитор, который инициирует процессы контроля, самодиагностирования и осуществляет реконфигурацию по результатам СД состояния компонентов очередной РК.

Требуется разработать такой метод контроля и СД компонентов ОУ-системы, чтобы после выполнения СД каждый исправный модуль имел одинаковую диагностическую информацию о состоянии отказавших компонентов очередной РК системы, включая компоненты, отказавшие в процессе СД.

МЕТОД САМОДИАГНОСТИРОВАНИЯ ОТКАЗАВШИХ КОМПОНЕНТОВ

Модель БГМ и связанные с ней правила дешифрации результатов тестирования предполагают, что в процессе тестирования компонента его техническое состояние (исправное или неисправное) остается неизменным. Однако эта предпосылка может и не выполняться во время процесса СД. Но если отказ компонента возникнет в процессе СД, то одни МПЩ будут тестировать его в исправном состоянии, а другие — в неисправном состоянии. Тогда дешифрация результатов тестирования может дать неправильный диагноз. Для обнаружения отказов, возникающих в процессе СД, и получения правильного диагноза будем дважды болеекратно запускать процесс СД и сравнивать результаты двух соседних запусков. Совпадение результатов первого и второго запусков свидетельствует о том, что неисправности компонентов, подозреваемых в отказе, имели место до первого запуска, а потому не привели к расхождению полу-

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



ченных результатов тестирования. Если же эти результаты не совпали, то в процессе первого или второго запусков возникли новые неисправности компонентов системы. В таком случае процесс СД нужно продолжить и повторять его до тех пор, пока не совпадут результаты проверок двух соседних запусков: второго и третьего, третьего и четвертого и т. д. Если учесть, что в высоконадежных системах неисправности компонентов бывают сравнительно редкими, то с большой вероятностью можно считать, что в течение двух—четырёх запусков СД может быть не больше одного отказа. Эта особенность возникновения и проявления одиночного отказа и заложена в четырехкратном (максимум) повторении процесса СД. Отметим, что существует ненулевая (хотя и очень малая) вероятность того, что новый отказ возникнет в модуле в тот интервал времени, когда он исполняет функции проверяющего для своих преемников. В этом случае, скорее всего, процесс СД будет прерван, и спустя некоторое время диагностический монитор запустит процесс контроля вновь, но уже с другого начального модуля.

Контроль цифровой системы состоит в том, что каждый исправный модуль должен проверить состояние модулей, непосредственно с ним связанных (преемников), и соответствующих линий связи. Для оптимизации последовательности выполнения проверок одного модуля разными МПЩ на диагностическом графе системы построим гамильтонов цикл. Для излагаемого ниже метода достаточно взять любой гамильтонов цикл, который обходит все вершины диагностического графа РК исправной системы. Такой цикл выбирается один раз при проектировании системы контроля.

Как и в работе [6], будем представлять информацию о составе выбранной РК № w системы 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 $0 \leq i_w, \dots, n_w \leq n + m - 1$) записаны номера

модулей-преемников m_{k1}, m_{k2}, \dots ; в ней также есть поля для записи результатов проверки модулем m_j этих преемников $r_{j, k1}, r_{j, k2}, \dots$ и значений их ранга при ранжировании графа. Пример такой таблицы T_{S0} , построенной для РК № 0 (см. табл. 1), показан в табл. 2. Как видно, в каждой из ее восьми строк заполнены только клетки для проверяющего модуля m_j и его преемников — m_{k1}, \dots, m_{k3} .

Для отображения взаимного положения модулей в гамильтоновом цикле будем записывать в строке T_j преемников модуля m_j так, что модуль m_{k1} является соседом справа модуля m_j , который, в свою очередь, является соседом справа для своего последнего преемника (модуля m_{k3} в рассматриваемом примере). 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 .

Традиционно процесс контроля запускается диагностическим монитором через определенные интервалы времени, что предотвращает накопление отказавших компонентов в системе. Назначим начальной вершиной процесса контроля одну из вершин гамильтонова цикла в выбранной РК № w , 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 $r_{n, k1}, r_{n, k2}, \dots$ строки T_n записать результаты проверок в двоичном коде: 0 — исправен, 1 — подозревается в отказе;
- оценить результаты проверок:
 - если результаты всех проверок — нулевые ($r_{n, k1} = 0, r_{n, k2} = 0, \dots$), то передать модулю m_{k1} управление продолжением процесса контроля;

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	—	—

- если среди полученных результатов проверок есть как единичные, так и нулевые (например, $r_{n, kj} = 0$), то передать одному из модулей с нулевым результатом (модулю m_{kj}) управление для перехода к процессу СД;
- 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_{k1} и каждый из последующих модулей выполняют 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 контроля получен единичный результат, свидетельствующий об обнаружении одного или нескольких отказавших компонентов, то процесс контроля прерывается, и диагностический монитор переходит к процессу СД.

На основе информации, полученной в процессе контроля, монитор начинает первый запуск процесса СД с того модуля m_j , исправность которого подтверждена проверкой, выполненной тем же модулем m_j , который первым получил единичный результат при проверке другого преемника — модуля m_k . Начальный модуль m_j ($\neq m_n, m_{n2}$) должен выполнить следующие действия:

- из таблицы состава T_{Sw} выделить строку T_j , переписать ее в промежуточную таблицу PT_{j1} (в индексе символа указан номер j начального МПЩ и номер запуска СД);
- в строке T_j найти номера преемников m_{k1}, m_{k2}, \dots ; проверить их состояние и в клетки $r_{j, k1}, r_{j, k2}, \dots$ строки T_j записать результаты проверок;
- по нулевым результатам проверок в строке T_j выделить подмножество номеров исправных преемников $\{q_j\}$ и первому из них передать таблицу PT_{j1} 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 или СД с другого начального модуля.

Очередной исправный модуль m_u , получив управление процессом СД, должен выполнить следующие действия (функции МПЩ):

1) из таблицы состава T_{Sw} выделить строку T_u , переписать ее в таблицу PT_{j1} , полученную от предшественника m_j ($m_u - 1$);

2) в строке T_u найти номера преемников m_{k1}, m_{k2}, \dots ; проверить их состояние и в клетки $r_{u, k1}, r_{u, k2}, \dots$ 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_{j1} :

- если один из номеров подмножества $\{q_u\}$ отсутствует в списке номеров левого столбца PT_{j1} , то передать соответствующему модулю (m_{u+1}) таблицу PT_{j1} 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_{j1} , то одному из модулей с этими номерами (например, m_{k1}) передать таблицу PT_{j1} и функции МПЩ на следующем шаге;
- если все номера $\{q_u\}$ — в списке номеров левого столбца PT_{j1} , то перейти к п. 4;

4) найти кратчайший путь от модуля m_u к одному из исправных модулей $m_{u+1}, m_{u+2}, \dots, m_{u+v1}$, еще не исполнявших функции МПЩ, с помощью процедуры ранжирования (см. работы [6, 10]), передать найденному модулю m_{u+1} (m_{u+2}, \dots) таблицу PT_{j1} и функции МПЩ на следующем шаге; если при выполнении процедуры ранжирования выяснится, что все обнаруженные исправные модули исполняли функции МПЩ, то это означает, что первый запуск СД завершен.

Если данный запуск СД был первым, то для подтверждения результатов проверок, накопленных в таблице PT_{j1} , и обнаружения возможных новых отказов диагностический монитор должен выполнить второй запуск СД с модуля m_{u+v1} , на котором закончился первый запуск.

Для совпадения описаний процедур, выполняемых в первом и втором запусках, модулю m_{u+v1} присвоим тот же символ m_j (начальный модуль второго запуска СД). Тогда модуль m_j должен выполнить такие действия:

- образовать новую таблицу PT_{j2} и переписать в нее из таблицы PT_{j1} строку T_{u+v1} с результатами проверок, полученными модулем m_{u+v1} в процессе первого запуска СД (в индексе символа данной таблицы указан номер j начального МПЩ и номер запуска);
- по нулевым результатам проверок в строке T_j ($\equiv T_{u+v1}$) выделить подмножество номеров исправных преемников $\{q_j\}$ и первому из них (например, модулю m_u) передать таблицы PT_{j1}, PT_{j2} и функции МПЩ на следующем



шаге СД. (В отличие от первого запуска такой исправный преемник всегда существует, так как модуль m_{u+v1} ($\equiv m_j$) получал управление в первом запуске процесса СД от исправного модуля.)

Очередной исправный МПЩ m_u , получив управление процессом СД, должен выполнить те же действия, что и в первом запуске, за исключением того, что результаты всех проверок записываются в таблицу PT_{j2} , а не в таблицу PT_{j1} , содержимое которой сохраняется неизменным на протяжении второго запуска.

В процессе выполнения переходов по цепочке исправных модулей во втором запуске СД рано или поздно выяснится, что все обнаруженные исправные модули уже исполнили функции МПЩ, т. е. процедуры второго запуска СД закончены. Тогда модуль m_{u+v2} , у которого во втором запуске совпали списки исправных и проверяющих модулей, должен сравнить содержимое таблиц PT_{j1} и PT_{j2} : сначала номера строк, затем результаты проверок $r_{j1,k}$, $r_{j2,k}$, ..., записанные в одноименных строках этих таблиц:

- в случае несовпадения — запустить процесс СД в третий (четвертый) раз, используя в качестве начального модуль m_{u+v2} (m_{u+v3}) и в качестве начальной таблицу PT_{j2} (PT_{j3});

- а при совпадении записей — передать в обратном направлении таблицу PT_{j2} (PT_{j3} , PT_{j4}) каждому из исправных модулей, перечисленных в ее левом столбце.

Для обработки информации из полученной таблицы PT_{j2} (PT_{j3} , PT_{j4}) 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 $r_{j1,k}$, $r_{j2,k}$, $r_{j3,k}$. Для примера в табл. 3 показаны две такие таблицы, построенные для РК № 0 (см. табл. 1), содержащие по восемь строк. Они различаются записями в столбцах, озаглавленных “Результаты проверок” и “Диагностическое заключение”. Эти записи появляются в диагностической таблице после проведения СД.

Нетрудно заметить, что при $j = k$ преемники модуля m_j , показанные в строке T_j табл. 2, совпадают с предшественниками модуля m_k , приведенными в левой части строки D_k табл. 3, а и 3, б. Это

Таблица 3

Диагностические таблицы 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	$a_{j1,k}$	$a_{j2,k}$	$a_{j3,k}$	
а. Диагностическая таблица T_{D0} для таблицы PT_{53}												
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	
б. Диагностическая таблица T_{D0}^1 для таблицы PT_{31}												
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	0	1	0	1	0	1	

совпадение — не случайно, оно следует из того, что в системе используются двунаправленные связи, поэтому модули m_{k1} , m_{k2} , m_{k3} — преемники модуля m_j — являются также его предшественниками m_{j1} , m_{j2} , m_{j3} .

Процедура обработки исправным модулем информации из таблицы PT_{j2} (PT_{j3} , PT_{j4}) заключается в следующем:

1) из строк T_j , ..., T_u , ..., T_{u+vi} ($i = 2, 3, 4$) таблицы PT_{j2} (PT_{j3} , PT_{j4}) выделить результаты проверок $r_{j,k1}$, ..., $r_{u,k1}$, ..., $r_{u+vi,k1}$ и записать их в соответствующие клетки строк T_D таблицы T_{Dw} ; в оставшихся клетках этих столбцов сделать прочерки (—), так как результатов этих проверок нет;

2) проанализировать результаты проверок, содержащиеся в строке D_k таблицы T_{Dw} , и записать в клетки s_k , $a_{j1,k}$, $a_{j2,k}$, ... этой строки символы диагностического заключения, полученные по следующим правилам:

- если в клетках результатов проверок строки D_k есть хотя бы один нуль (например, $r_{j1,k} = 0$), то записать 0 в клетки на пересечениях строки D_k со столбцами s_k и $a_{j1,k}$;
- если в этой же строке D_k есть еще 0 (1), то записать 0 (1) в клетки на пересечениях строки D_k со столбцами переменных соответствующих линий;
- если в клетках результатов проверок строки D_k записаны только единицы, то записать 1 в клетку на пересечении строки D_k со столбцом s_k и записать буквы z в клетки на пересечениях строки D_k со столбцами переменных соответствующих линий;
- в оставшихся клетках этих столбцов сделать прочерки (—).

Теперь расшифровка информации из таблицы PT_{j2} (PT_{j3} , PT_{j4}) сводится к замене символов 0, 1, (—) и z , записанных в клетках строки D_k таблицы T_{Dw} , на заключения о состояниях модуля m_k и инцидентных линий связи согласно правилам: 0 → “модуль исправен”, “линия исправна”; 1 → “модуль (линия) подозревается в отказе”; (—) неопределенное состояние линии связи, потому что соответствующая проверка не выполнялась; z → неопределенное состояние линии связи, так как отказавший модуль “маскирует” состояние инцидентной линии связи.

Поскольку каждый исправный модуль получил одинаковую информацию в виде таблицы PT_{j2} (PT_{j3} , PT_{j4}), то после ее обработки все эти модули будут иметь одинаковые диагностические заключения о состоянии компонентов системы.

ПРОЦЕДУРА РЕКОНФИГУРАЦИИ

Обработка результатов проверок из таблицы PT_{j2} (PT_{j3} , PT_{j4}) определяет подмножество компонентов в составе РК № w , подозреваемых в отказе. Процедура реконфигурации m -ОУ системы сводится к нахождению в первичной таблице (или в одной из вторичных таблиц) такой РК № w^* , в состав которой не входят компоненты, подозреваемые в отказе: т. е. отказавшая вершина займет место одной из резервных, отказавшая дуга не будет соединять пару смежных рабочих вершин и не будет входить в состав дуг, перечисленных в правой части строки w^* .

После нахождения такой работоспособной РК № w^* диагностический монитор модифицирует таблицы T_{Sw} и T_{Dw} , заменив в них номера рабочих модулей прежней РК № w на соответствующие номера новой РК № w^* . Затем он 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 запустит процесс СД для идентификации тех компонентов, отказы которых нарушили работоспособность РК № w^* .

ПРИМЕР

Особенности процессов СД и устранения возникающих отказов покажем на примере той же 2-ОУ системы, построенной для целевой системы со структурой трехмерного куба, что и в работе [6]. Рассматривается 3-куб, в вершинах которого расположены процессорные элементы. На рис. 1 они обозначены кружками, в которых указаны их номера. Двукратное применение алгоритмов синтеза [2] к графу 3-куба G_0^8 (см. рис. 1) позволяет построить 2-ОУ граф G_R^{10} (см. рис. 2); подробности применения алгоритмов синтеза к графу G_0^8 можно найти в статье [5]. Первичная таблица РК графа G_R^{10} представлена табл. 1. Она состоит из 10-ти строк, соответствующих десяти РК и обозначенных номерами 0, 1; ..., 9; клетки столбцов 1, 2, ..., 8 каждой строки содержат номера рабочих вершин данной РК, клетки столбцов 9 и 10 — номера резервных вершин. В клетках столбцов 12—15 приведены номера дуг, соответствующие номерам четырех внутренних дуг целевого графа G_0^8 . Начальная РК № 0 показана в строке 0: рабочие вершины 0, 1, ..., 5, 4, резервные 8, 9, рабочие дуги (0—2), (3—7), (6—4), (5—1). В столбце 11 показана дуга (4—0), которая замыкает гамильтонов цикл из рабочих вершин: 0 → 1 → 3 → 2 → 6 → 7 → 5 → 4 → 0.



Дуги между смежными вершинами этого цикла в табл. 1 отдельно не показаны, так как их номера однозначно определяются 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 предыдущей строки $0, 1, \dots, 8$. Номера дуг, показанные в клетках столбцов 11—15 каждой строки, 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.

Отметим, что алгоритм А2 оптимального обхода и тестирования вершинных процессорных элементов [11] 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 применим описанный процесс последовательного контроля компонентов РК № 0 и тех РК, в которые переходит система для исключения отказавших компонентов.

Итак, пусть 2-ОУ система работает в начальной РК № 0 (см. табл. 1). Предположим, что между циклами контроля отказала линия связи $(5-1)$ (отмечена знаком * в клетках тех РК табл. 1, в состав которых она входит). Контроль работоспособности 2-ОУ системы начинается с модуля 0, соответствующего начальной вершине выбранного гамма-тона цикла для РК № 0.

По сигналу диагностического монитора начальный модуль m_0 выполнит такие действия: из таблицы T_{S0} состава РК № 0 (см. табл. 2) выделит строку T_0 , найдет в ней номера преемников m_{k1} ($\equiv 1$), m_{k2} ($\equiv 2$), m_{k3} ($\equiv 4$), проверит их состояние и в клетки $r_{j,k1}$, $r_{j,k2}$, $r_{j,k3}$ строки T_0 запишет результаты проверок: $r_{0,1} = 0$, $r_{0,2} = 0$, $r_{0,4} = 0$ (см. табл. 4); так как результаты всех проверок — нулевые, то передаст модулю 1 ($\equiv m_{k1}$) управление для продолжения контроля (здесь и в последующем полужирным шрифтом выделен тот нуль, по которому передается управление следующему МПЩ для продолжения контроля или СД).

Модуль 1, получив управление от модуля 0.

— 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);

— оценит результаты проверок и, обнаружив $r_{1,5} = 1$ и $r_{1,3} = 0$, передаст модулю 3 управление для перехода к процессу СД.

Начальный модуль 3 первого запуска процесса СД выполнит такие действия:

— из таблицы состава T_{S0} выделит строку T_3 , переписет ее в новую таблицу PT_{31} (в индексе символа таблицы указан номер 3 начального МПЩ для первого запуска СД);

— в строке T_3 найдет номера преемников 2, 7, 1; проверит их состояние и в соответствующие клетки строки T_3 запишет результаты проверок;

— по нулевым результатам проверок в строке T_3 выделит номера исправных преемников $\{q_3 = 2, 7, 1\}$ и первому из них, модулю 2, передаст таблицу PT_{31} и функции МПЩ на следующем шаге СД.

В свою очередь, модуль 2 выполнит такие действия:

1) из таблицы T_{S0} выделит строку T_2 и переписет ее в таблицу PT_{31} (см. табл. 4);

2) в строке T_2 найдет номера преемников 6, 0, 3; проверит их и в соответствующие клетки строки T_2 запишет результаты проверок;

3) по нулевым результатам проверок в строке T_2 выделит номера исправных преемников $\{q_2 = 6, 0, 3\}$, сравнит их с номерами 3, 2 в левом столбце таблицы PT_{31} и обнаружит, что модули 6 и 0 не исполняли функций МПЩ, поэтому передаст первому из них (модулю 6) таблицу PT_{31} с результатами проверок и функции МПЩ на следующем шаге СД, при этом модуль 6 вернется к п. 1 процесса СД.

Аналогичные действия в роли МПЩ поочередно выполняют модули 6, 7, результаты их проверок своих преемников показаны в строках T_6 , T_7 , табл. 4. Предположим, что после того, как модуль 6 проверил своих преемников и передал таблицу PT_{31} и управление модулю 7, модуль 4 отказал. Получив управление от модуля 7, модуль 5, выполняя функции МПЩ, проверит своих преемников и обнаружит, что его исправный преемник — модуль 7 уже исполнял функции МПЩ. Поэтому модуль 5 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_3, \dots, T_5 таблицы PT_{31} , модуль 5 определит, что функции МПЩ не исполняли исправные модули 1 и 0, из таблицы T_{S0} (см. табл. 2) выделит строки T_1

и T_0 , добавит их в таблицу PT_{31} . Затем, выполняя процедуру ранжирования, модуль 5 присвоит ранг 0 модулю 5, ранг 1 модулю 7, затем ранг 2 модулям 3 и 6, ранг 3 модулям 1 и 2. На этом ранжирование будет закончено, так как модуль 1 попал в число ранжированных. (Результаты ранжирования показаны в строках таблицы PT_{31} ; 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 из-за единичных результатов их проверок.) Теперь модуль 5 передаст таблицу PT_{31} и функции МПЩ на следующем шаге СД модулю 1 через цепочку модулей: $5 \rightarrow 7 \rightarrow 3 \rightarrow 1$.

Модуль 1, получив управление от модуля 5:

1) в строке T_1 таблицы PT_{31} (см. табл. 4) найдет номера преемников 3, 5, 0; проверит их и в соответствующие клетки строки T_1 запишет результаты проверок;

2) по нулевым результатам проверок в строке T_1 выделит номера исправных преемников $\{q_1 = 3, 0\}$ и передаст таблицу PT_{31} и функции МПЩ на следующем шаге СД модулю 0, который после исполнения функций МПЩ обнаружит, что исправных модулей, не исполнявших функции МПЩ, нет, поэтому первый запуск СД завершен.

Для подтверждения результатов проверок, накопленных в таблице PT_{31} , и обнаружения возможных новых отказов монитор должен запустить процесс СД второй раз с модуля 0, на котором за-

Таблица 4

Информация для РК № 0

Символ строки	Проверяющий модуль		Проверяемые модули, 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	0	—
T_1	1	—	3	0	—	5	1	—	0	0	—
Таблица PT_{31}											
T_3	3	2	2	0	3	7	0	1	1	0	3
T_2	2	3	6	0	4	0	0	4	3	0	2
T_6	6	2	7	0	1	4	0	3	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_1	1	3	3	0	—	5	1	—	0	0	—
T_0	0	—	1	0	—	2	0	—	4	1	—
Таблица PT_{02}											
T_0	0	—	1	0	—	2	0	—	4	1	—
T_1	1	—	3	0	—	5	1	—	0	0	—
T_3	3	—	2	0	—	7	0	—	1	0	—
T_2	2	—	6	0	—	0	0	—	3	0	—
T_6	6	—	7	0	—	4	1	—	2	0	—
T_7	7	—	5	0	—	3	0	—	6	0	—
T_5	5	—	4	1	—	1	1	—	7	0	—
Таблица PT_{53}											
T_5	5	—	4	1	—	1	1	—	7	0	—
T_7	7	—	5	0	—	3	0	—	6	0	—
T_0	6	—	7	0	—	4	1	—	2	0	—
T_2	2	—	6	0	—	0	0	—	3	0	—
T_3	3	—	2	0	—	7	0	—	1	0	—
T_1	1	—	3	0	—	5	1	—	0	0	—
T_0	0	—	1	0	—	2	0	—	4	1	—



кончился первый запуск. В начале второго запуска модуль θ выполнит такие действия:

- образует новую таблицу PT_{02} (см. табл. 4) и переписывает в нее из таблицы PT_{31} строку T_0 с результатами проверок, полученными в первом запуске СД (в индексе символа таблицы PT_{02} указан номер θ начального МПЩ во втором запуске);
- по нулевым результатам проверок в строке T_0 выделит номера исправных преемников $\{q_0 = 1, 2\}$ и передаст первому из них (модулю 1) таблицы PT_{31} , PT_{02} и функции МПЩ на следующем шаге СД.

Исправный модуль 1, а затем модули 3, 2, 6, 7, 5, получив управление процессом СД, выполняют те же действия, что и в первом запуске, но результаты всех проверок записывают в таблицу PT_{02} (см. табл. 4), а содержимое PT_{31} передают неизменным на протяжении второго запуска. Модуль 5 обнаружит, что исправных модулей, не исполнявших функции МПЩ, больше нет, поэтому второй запуск СД завершен.

Сравнив записи в таблицах PT_{31} и PT_{02} , модуль 5 определит их различие по результатам проверки модуля 4: в таблице PT_{31} $r_{6,4} = 0$, а в таблице PT_{02} $r_{6,4} = 1$; поэтому монитор запустит процесс СД в третий раз, используя в качестве начального модуль 5 и в качестве начальной таблицу PT_{02} .

Результаты проверок, полученных в третьем запуске СД с модуля 5, показаны в строках T_5, \dots, T_0 таблицы PT_{53} (в нижней части табл. 4). Особенность третьего запуска: не потребовалась процедура ранжирования, так как управление СД каждый раз передавалось последнему из исправных преемников очередного МПЩ. После завершения третьего запуска модуль θ , сравнив таблицы PT_{02} и PT_{53} , обнаружит их полное совпадение как по номерам строк, так и по результатам проверок в этих строках. Поэтому модуль θ передаст таблицу PT_{53} каждому из исправных модулей по цепочке: $\theta \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 5$.

Теперь каждый исправный модуль РК № 0 системы выполнит следующую обработку информации из таблицы PT_{53} : из строк T_5, \dots, T_0 выделит результаты проверок и запишет их в клетки строк T_{D0} (см. табл. 3, а); в соответствии с правилами, приведенными в основном тексте статьи, проанализирует результаты проверок, содержащиеся в каждой строке D_k таблицы T_{D0} , и запишет в клетки $s_k, a_{j1,k}, a_{j2,k} \dots$ этой же строки символы соответствующего диагностического заключения.

Расшифровка символов, показанных в клетках столбцов “Диагностическое заключение” таблицы T_{D0} , позволяет прийти к такому диагнозу:

- подозреваются в отказе модуль 4 и линии связи (5—1) и (1—5);
- нет информации о состоянии трех линий, отмеченных прочерками (—): отказавший модуль 4 не использовался в качестве МПЩ;
- нет информации о состоянии трех линий, отмеченных символом z в строке D_4 , так как отказавший модуль 4 “маскирует” отказы своих входных линий;
- в РК № 0 исправны 7 модулей ($\theta, 1, 2, 3, 5, 6, 7$) и 16 линий связи, соответствующих тем клеткам в столбцах $a_{j1,k}, \dots, a_{j3,k}$, в которых стоят нули.

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

В табл. 3, б показана диагностическая таблица T_{D0}^1 , построенная для информации из таблицы PT_{31} . Сравнение диагностических заключений, приведенных в табл. 3, а и 3, б, показывает различие записей в строке D_4 : если в таблице T_{D0} модуль 4 подозревается в отказе, то в таблице T_{D0}^1 модуль 4 признан исправным, а подозреваются в отказе линии (0—4) и (5—4). Причина такого диагноза в том, что по результату проверки модуля 4, выполненной модулем 6 до его отказа, он был признан (естественно) исправным, а единичные результаты его последующих проверок модулями 5 и θ (после отказа) были отнесены к отказам линий (5—4) и (0—4). Таким образом, если бы процесс СД был ограничен одним запуском, то получили бы неправильный диагноз.

Для устранения подозреваемых компонентов: модуля 4 и дуг (5—1), (1—5) диагностический монитор должен перевести 2-ОУ систему из РК № 0 в такую РК, в которую не входят эти компоненты. Обратившись к первичной таблице (см. табл. 1), нетрудно обнаружить, что модуль 4 входит в состав восьми РК — от РК № 0 до РК № 7 — и не входит только в РК № 8 и 9, где он записан в числе резервных модулей; в эти же РК не входит и дуга (5—1). Для определенности, в качестве следующей выберем РК № 8. На рис. 2 представлен граф РК № 8: рабочие вершины и дуги показаны толстыми линиями, резервная вершина 5 и резервные дуги — тонкими линиями, а неисправная вершина 4 и пара дуг (5—1) и (1—5) — тонкими пунктирными линиями.

После смены РК для РК № 8 получим такой гамильтонов цикл: $8 \rightarrow 9 \rightarrow \theta \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 8$. С учетом этого цикла монитор модифицирует таблицы T_{S0} и T_{D0} , заменит номера рабочих модулей в соответствии с подстановкой (89013267)

вместо (01326754) и получит таблицу T_{S8} (табл. 5) и диагностическую таблицу T_{D8} (табл. 6). Затем монитор запустит процесс контроля РК № 8, который подтвердит ее работоспособность.

Несколько слов об устранении компонентов, показанных в диагностической таблице T_{D0}^1 таблицы для PT_{31} (см. табл. 3, б). Как видно из первичной табл. 1, есть только три РК: №№ 7, 8 и 9, в состав которых не входят подозреваемые линии (5—1), (1—5), (0—4) и (5—4). Если монитор выберет РК № 8, то функционирование системы совпадет с описанным. Если монитор выберет РК № 7, то контроль признает ее неработоспособной, так как среди рабочих модулей есть отказавший модуль 4, и запустит процесс СД. На основании нового диагноза, который будет правильным, монитор переведет систему в РК № 8.

Несколько иная ситуация будет при переходе в РК № 9: действительно отказавший модуль 4 попадет на место резервной вершины (в столбце 9), и последующий контроль признает РК № 9 работоспособной. Однако, поставив на место резерв-

ного отказавший модуль 4, мы фактически оставили систему без резерва. Поэтому система продолжит рабочее функционирование до первого отказа любого компонента РК № 9. Таким образом, учет неправильного диагноза приведет к преждевременному и неэффективному расходованию резервных компонентов и снижению времени надежного функционирования системы.

Если бы спустя некоторое время работы в РК № 8 возникли отказы модуля 3 и линии (6—9), то последующие действия системы по организации контроля, СД и реконфигурации можно найти в работе [6]. Здесь же будем считать, что между соседними включениями системы контроля произошел отказ линии (8—1), а затем в процессе первого запуска СД между проверками модуля 1 модулем 3 и модуля 3 модулем 2 произошел отказ линии (2—3) (эти линии показаны на рис. 2 толстым пунктиром).

Итак, по сигналу диагностического монитора модуль 8 начнет контроль работоспособности РК № 8: выполнит действия, аналогичные действиям модуля 0 в РК № 0, но искомым строку T_8 возьмет

Таблица 5

 Таблица состава T_{S8} для РК № 8

Символ строки	Проверяющий модуль и ранг		Преимники 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 таблицы PT_{23}

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	1	0	0	0	1	0
D_9	0	6	8	9	0	0	0	0	0	0	0
D_0	1	2	9	0	0	0	0	0	0	0	0
D_1	3	8	0	1	0	1	0	0	0	1	0
D_3	2	7	1	3	1	0	0	0	1	0	0
D_2	6	0	3	2	0	0	1	0	0	0	1
D_6	7	9	2	6	0	0	0	0	0	0	0
D_7	8	3	6	7	0	0	0	0	0	0	0



из таблицы TS_8 (см. табл. 5) (результаты проверок, выполненных модулем 8, показаны в строке T_8 табл. 7); обнаружив $r_{8,1} = 1$ и $r_{8,9} = 0$, модуль 8 переvedет монитор в режим СД.

Начальный модуль 9 первого запуска СД: из таблицы T_{S8} (см. табл. 5) выделит строку T_9 , переписывает ее в промежуточную таблицу PT_{91} ; в строке T_9 найдет номера приемников 0, 6, 8; проверит их состояние и в соответствующие клетки строки T_9 запишет результаты проверок; затем по нулевым результатам проверок выделит номера исправных приемников $\{q_9 = 0, 6, 8\}$ и первому из них, модулю 0, передаст таблицу PT_{91} и функции МПЩ на

следующем шаге СД (результаты проверок, выполненных модулем 9, показаны в строке T_9 таблицы PT_{91} (см. табл. 7)).

Действия модулей 0, 1, ..., 8 подобны действиям МПЩ, ранее описанным в процессе СД в РК № 0, за исключением того, что необходимые строки будут извлекаться из табл. T_{S8} (см. табл. 5); а результаты выполненных ими проверок записываться в строки таблицы PT_{91} (см. табл. 7). Отметим, что после того, как модуль 3 передал таблицу PT_{91} и функции МПЩ модулю 2, отказала линия (2—3), что нашло отражение в результате проверки $r_{2,3} = 1$. В конце запуска модуль 8, выполняя функции

Таблица 7

Информация для РК № 8

Символ строки	Проверяющий модуль и ранг		Преемники m_p , 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 № 8											
T_8	8	—	9	0	—	1	1	—	7	0	—
Таблица PT_{91}											
T_9	9	—	0	0	—	6	0	—	8	0	—
T_0	0	—	1	0	—	2	0	—	9	0	—
T_1	1	—	3	0	—	8	1	—	0	0	—
T_3	3	—	2	0	—	7	0	—	1	0	—
T_2	2	—	6	0	—	0	0	—	3	1	—
T_6	6	—	7	0	—	9	0	—	2	0	—
T_7	7	—	8	0	—	3	0	—	6	0	—
T_8	8	—	9	0	—	1	1	—	7	0	—
Таблица PT_{82}											
T_8	8	—	9	0	—	1	1	—	7	0	—
T_9	9	—	0	0	—	6	0	—	8	0	—
T_0	0	—	1	0	—	2	0	—	9	0	—
T_1	1	—	3	0	—	8	1	—	0	0	—
T_3	3	—	2	1	—	7	0	—	1	0	—
T_7	7	—	8	0	—	3	0	—	6	0	—
T_6	6	—	7	0	—	9	0	—	2	0	—
T_2	2	—	6	0	—	0	0	—	3	1	—
Таблица PT_{23}											
T_2	2	—	6	0	—	0	0	—	3	1	—
T_6	6	—	7	0	—	9	0	—	2	0	—
T_7	7	—	8	0	—	3	0	—	6	0	—
T_8	8	—	9	0	—	1	1	—	7	0	—
T_9	9	—	0	0	—	6	0	—	8	0	—
T_0	0	—	1	0	—	2	0	—	9	0	—
T_1	1	—	3	0	—	8	1	—	0	0	—
T_3	3	—	2	1	—	7	0	—	1	0	—

МПЩ, проверит своих преемников и обнаружит, что его исправные преемники (модули 9 и 7) уже исполняли функции МПЩ, перейдет к процедуре ранжирования, в начале которой обнаружит, что больше нет исправных модулей для исполнения функций МПЩ; таким образом, первый запуск СД завершен.

В начале второго запуска СД модуль 8, на котором закончился первый запуск, образует таблицу PT_{82} (см. табл. 7), переписывает в нее из таблицы PT_{91} строку T_8 с результатами проверок, полученными в первом запуске СД, по нулевым результатам проверок в строке T_8 выделяет номера исправных преемников $\{q_8 = 9, 7\}$ и передаст первому из них, модулю 9, таблицы PT_{91} , PT_{82} и функции МПЩ на следующем шаге СД.

Исправный модуль 9, а затем модули 0, 1, ..., 6, 2, получив управление процессом СД, выполняя те же действия, что и в первом запуске, но результаты всех проверок записывают в таблицу PT_{82} (см. табл. 7), а содержимое таблицы PT_{91} передают неизменным на протяжении второго запуска; номер модуля, которому передаются функции МПЩ на следующем шаге, отмечен в таблицу PT_{82} полужирным нулем в соседней клетке результата проверки. Модуль 2 обнаружит, что исправных модулей, не исполнявших функции МПЩ, больше нет, поэтому второй запуск СД завершен.

Сравнив записи в таблицах PT_{91} и PT_{82} , модуль 2 определит их различие по результатам проверки $r_{3,2}$: в таблице PT_{91} $r_{3,2} = 0$, а в таблице PT_{82} $r_{3,2} = 1$; поэтому диагностический монитор запустит процесс СД в третий раз, используя в качестве начального модуль 2 и в качестве начальной таблицу PT_{82} .

Результаты проверок, полученных в третьем запуске СД с модуля 2, показаны в строках T_2, \dots, T_3 таблицы PT_{23} (в нижней части табл. 7). После за-

вершения третьего запуска модуль 3, сравнив таблицы PT_{82} и PT_{23} , обнаружит их полное совпадение и передаст таблицу PT_{23} каждому из исправных модулей по цепочке $3 \rightarrow 1 \rightarrow 0 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 2$.

Затем каждый исправный модуль РК № 8, получив таблицу PT_{23} , выделит из ее строк результаты проверок и запишет их в клетки строк таблицы T_{D8} (см. табл. 6); в соответствии с правилами, приведенными в основном тексте статьи, проанализирует их и запишет в клетки $s_k, a_{j1,k}, a_{j2,k}, \dots$ каждой строки D_k соответствующие символы. Их расшифровка приводит к следующим диагностическим заключениям:

- подозреваются в отказе пары линий связи (1—8) и (8—1), (2—3) и (3—2);
- в РК № 8 исправны все восемь модулей и остальные линии связи, кроме перечисленных.

Сравнение полученного диагноза с составом исходной отказовой ситуации показывает, что вместо линий (2—3) и (8—1) подозреваются в отказе указанные пары линий; причина та же, что и при диагностировании линии (5—1) в РК № 0.

Для исключения подозреваемых компонентов из состава РК диагностический монитор должен перевести 2-ОУ систему в такую РК, в которую не входят эти компоненты, а также ранее отказавшие модуль 4 и линии (5—1), (1—5). Для поиска такой РК целесообразно воспользоваться вторичной таблицей, построенной на основании РК № 8. Такая таблица представлена в табл. 8 и отличается от первичной таблицы (см. табл. 1) тем, что имеет только один столбец (№ 9) для резервной вершины, а столбец 10 закреплен за отказавшим модулем 4. Отметим в клетках табл. 9 подозреваемые компоненты: линию (8—1) знаком *, а другую линию (2—3), связывающую соседние рабочие вершины, знаками (—), стоящими после цифры 3 и перед цифрой 2. Теперь поиск работоспособной РК сводится к нахождению в табл. 8 такой строки,

Таблица 8

Вторичная таблица 1

№№ п/п	Номера рабочих вершин								Резервная вершина	Отказавшая вершина	Номера дуг				
	1	2	3	4	5	6	7	8			9	10	11	12	13
8_0	8	9	0	1	3—	—2	6	7	5	4	7—8	8—1*	0—2	3—7	6—9
8_1	9	0	1	3—	—2	6	7	5	8	4	5—9	9—3	1—6	2—5	7—0
8_2	0	1	3—	—2	6	7	5	8	9	4	8—0	0—2	3—7	6—8	5—1
8_3	1	3—	—2	6	7	5	8	9	0	4	9—1	1—6	2—5	7—9	8—3
8_4	3—	—2	6	7	5	8	9	0	1	4	0—3	3—7	6—8	5—0	9—2
8_5	—2	6	7	5	8	9	0	1	3—	4	1—2	2—5	7—9	8—1*	0—6
8_6	6	7	5	8	9	0	1	3—	—2	4	3—6	6—8	5—0	9—3	1—7
8_7	7	5	8	9	0	1	3—	—2	6	4	2—7	7—9	8—1*	0—2	3—5
8_8	5	8	9	0	1	3—	—2	6	7	4	6—5	5—0	9—3	1—6	2—8



в которую не входят подозреваемые компоненты. Анализ записей табл. 8 показывает, что только одна строка с номером 8_6 удовлетворяет этим требованиям: в ней модуль 2 оказался в числе резервных, а линия ($8-1$) не входит в число рабочих.

После смены РК № 8 на РК № 8_6 получим такой гамильтонов цикл: $6 \rightarrow 7 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 6$. С учетом этого цикла диагностический монитор модифицирует таблицы T_{S8} и T_{D8} , заменит номера рабочих модулей в соответствии с подстановкой (67589013) вместо (89013267), модифицирует таблицу состава и диагностическую таблицу, запустит процесс контроля РК № 8_6 , который подтвердит ее работоспособность.

Для оценки эффективности реконфигурации 2-ОУ системы с целью устранения новых отказов в РК № 8_6 нужно построить вторичную таблицу на основании строки 8_6 из табл. 8. (Для сокращения объема статьи она здесь не приводится.) Анализ ее строк показывает, что из восьми РК только четыре работоспособны, но включают в себя одни и те же рабочие вершины и рабочие дуги. Другими словами, все эти РК соответствуют одной и той же избыточной восьмивершинной подсистеме, получающейся из 2-ОУ системы после удаления отказавших компонентов: модуля 4 и линий ($1-5$), ($5-1$), ($8-1$), ($1-8$), ($2-3$), ($3-2$). Следовательно, новый отказ любого компонента, входящего в состав РК № 8_6 , приведет к отказу всей системы и потребует вмешательства человека-оператора для устранения накопившихся неисправностей и восстановления работоспособности исходной системы.

Сравнение состава отказавших компонентов данного примера с оценкой (1) позволяет заключить, что отказы внутренних дуг ($1-5$) и ($8-1$) не препятствовали устранению отказов модуля 4 и внешней дуги ($2-3$) и не понизили эту оценку.

ЗАКЛЮЧЕНИЕ

Разработан метод самодиагностирования возникающих отказов компонентов (модулей и линий связи) цифровых систем с реконфигурацией, для построения графов структур которых применены алгоритмы синтеза, предложенные М.Ф. Караваем. В основу метода положен принцип расширяющихся областей: выполнение необходимых процедур начинается с исправного модуля и последовательно передается другим модулям, исправность которых подтверждена предшествующими проверками. Допускаются устойчивые кратные отказы ограниченного числа компонентов, во время диагностирования могут возникать одиночные отказы. Для обнаружения возникающих отказов и получения правильного диагноза применяется дву- и более кратный запуск процесса самодиагностирования со сравнением результатов двух последних запусков.

Достоверность полученных результатов диагностирования основана на том, что все процедуры самодиагностирования выполняются модулями, исправность которых подтверждена на предыдущих этапах. В связи с этим наиболее “вредными” для предложенного метода самодиагностирования являются те неисправности, которые проявляются у модуля, признанного ранее исправным и получившего управление для выполнения диагностических функций. Результаты работы неисправного модуля могут быть самыми разными: начиная от полной остановки процесса самодиагностирования и кончая “хитрыми” искажениями информации, не обнаруживаемыми принятой схемой контроля. Так как в современных системах вероятность проявления подобных неисправностей в специально подобранные и сравнительно небольшие интервалы времени достаточно мала, то можно заключить, что предложенный метод самодиагностирования будет давать правильный диагноз для подавляющего числа неисправных компонентов. И только в единичных случаях для устранения возникшей неисправности потребуются помощь человека-оператора.

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 multiprocessor 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. Ведешенков В. А. Самодиагностирование отказов цифровых систем с реконфигурацией // Проблемы управления. — 2003. — № 4. — С. 39—51.
7. Barsi F., Grandoni F., Maestrini P. A theory of diagnosability of digital systems // IEEE Trans. Comput. — 1976. — Vol. C-25. — № 6. — P. 585—593.
8. 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.
9. Пархоменко П. П. Определение технического состояния многопроцессорных вычислительных систем путем анализа графа синдромов // Автоматика и телемеханика. — 1999. — № 5. — С. 126—135.
10. Ведешенков В. А. Организация самодиагностирования технического состояния цифровых систем // Автоматика и телемеханика. — 2003. — № 11. — С. 165—182.
11. Пархоменко П. П. О системном диагностировании вершинных и реберных гиперкубовых структур многопроцессорных вычислительных систем // Автоматика и телемеханика. — 1996. — № 11. — С. 165—174.

☎ (095) 334-75-90

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

