

СПОСОБ САМОДИАГНОСТИРОВАНИЯ НЕОДНОРОДНЫХ ЦИФРОВЫХ СИСТЕМ

В. А. Ведешенков

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

Предложен способ самодиагностирования (СД) компонентов неоднородных цифровых систем, содержащих одинаковое число устройств двух типов (процессоров, памяти). Для осуществления контроля и СД связанные разнотипные устройства объединяются в подсистемы: проверяющие и проверяемые. Допускаются устойчивые кратные отказы ограниченного числа компонентов. Разработана организация СД рассматриваемых неоднородных цифровых систем, основанная на постепенном расширении множества исправных модулей. Дан пример реализации СД компонентов в 14-модульной системе.

ВВЕДЕНИЕ

Любой метод системного диагностирования технического состояния компонентов в цифровых системах (многопроцессорных и многомашинных вычислительных системах (ММВС)) основан на том, что вводится ограничение на максимальное число произвольно расположенных отказавших компонентов, которое позволяет оставшимся исправным модулям получить информацию, достаточную для правильного диагностирования состояния компонентов. Для описания результатов тестирования модулей в данной работе используется модель Барси—Грандони—Маестрини (БГМ) [1], которая предполагает, что модули, на которые разбивается исходная ММВС, однородные и обладают функциональными возможностями, достаточными для реализации диагностических алгоритмов, составными частями которых являются процедуры проверки компонентов системы и сбора результатов этих проверок.

Однако многие ММВС содержат устройства различных типов (процессоры, станции памяти, каналы связи). Примером такой ММВС может служить неоднородная система обработки радиолокационной информации [2], содержащая семь процессоров и семь станций памяти, соединенных каналами связи в соответствии с графом, показанным на рис. 1, где процессоры обозначены четными номерами, а станции памяти — нечетными. Любой процессор системы связан с любым другим процессором через одну станцию памяти. Наличие устройств разных типов не позволяет непосредственно применить системные методы диагностирования. Но автоматизированное диагностирование состояния компонентов не-

однородных цифровых систем представляется необходимым, так как позволяет сократить время восстановления работоспособности систем после обнаружения факта их неправильного функционирования.

Далее представлен способ организации контроля и самодиагностирования (СД) состояния компонентов неоднородных цифровых систем, предполагающий на

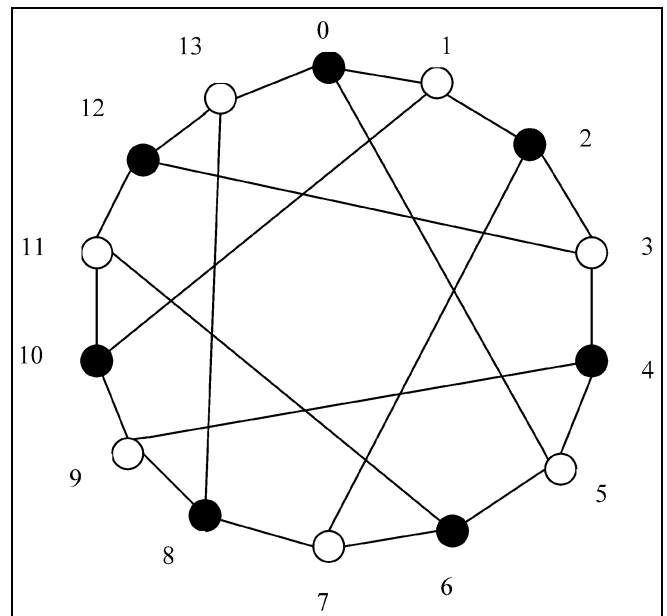


Рис. 1. Граф неоднородной системы $Q_{14, 21}$

этапах контроля и СД объединение связанных разнотипных устройств в проверяющие и проверяемые подсистемы, обладающие необходимыми функциональными возможностями для реализации диагностических алгоритмов.

Напомним, что в однородных системах при выполнении процессов контроля и СД проверяющий модуль подает на входы проверяемого модуля последовательность проверок (тест), получает ответные реакции проверяемого модуля на этот тест, сравнивает их с эталонными, а результат сравнения оценивает в двоичной форме: 0 — результаты правильные, 1 — неправильные. В модели БГМ предполагается, что оценка результатов теста всегда достоверна, если исправен проверяющий модуль; эта оценка равна 1, если неисправны оба модуля, и недостоверна (0 или 1) в том случае, когда неисправен проверяющий модуль, а проверяемый исправен. Поэтому нулевая оценка, выдаваемая проверяющим модулем, с большой вероятностью подтверждает исправность проверенного им модуля. Такая предпосылка модели БГМ существенно упрощает процедуру диагностирования, так как для идентификации исправных модулей достаточно 0-путей единичной длины (в терминологии работы [3]).

Суммарная оценка результатов выполненных тестов t_1, \dots, t_l образует двоичный вектор $R = r_1, \dots, r_l$, который называется реальным синдромом. В большинстве алгоритмов системного диагностирования предполагается (или констатируется), что результаты тестов, образующие реальный синдром R , собираются и обрабатываются (дешифрируются) внешним устройством. По результатам работы этого устройства (результатам диагностирования) обслуживающий персонал заменяет отказавшие компоненты и восстанавливает работоспособность ММВС. В настоящей статье, как и в работах [4, 5], развивается другой подход, основанный на организации самодиагностирования системы, когда необходимые алгоритмы, включая сбор реального синдрома R и его обработку, реализуются на исправных процессорах самой системы.

1. ПОСТАНОВКА ЗАДАЧИ

Пусть анализируемая система представлена неоднородным диагностическим графом $Q_{N, L}$, $N = 2n$ вершин которого отображают n активных устройств (процессоров) и n пассивных устройств (памятей) системы, а L ненаправленных дуг между вершинами — линии связи между активными и пассивными устройствами (процессорами и памятьями). Будем считать, что каждая вершина одного типа связана ненаправленными дугами с p вершинами другого типа, тогда $L = pn$.

Будем считать, что допускаются устойчивые отказы ограниченного числа вершин и дуг, причем их неисправности таковы, что прекращают работу модулей, в которых они возникают, и не влияют на работоспособность смежных модулей. Предполагается также, что во время выполнения процесса СД новые отказы не возникают. Для описания результатов тестирования компонентов применяется модель БГМ. Предполагается, что в исходной системе есть диагностический монитор, который организует процессы контроля и СД.

Требуется разработать такой способ организации контроля и СД компонентов неоднородной системы, чтобы после окончания СД каждое исправное активное устройство (процессор) имело одинаковую информацию для идентификации состояния остальных компонентов (процессоров, памяти и линий связи) системы.

2. СПОСОБ САМОДИАГНОСТИРОВАНИЯ НЕОДНОРОДНЫХ ЦИФРОВЫХ СИСТЕМ

Процесс СД компонентов однородных систем включает в себя [4, 5]:

- исполнение исправным модулем функций проверяющего;
- нахождение пути от одного исправного модуля к другому исправному для передачи ему функций проверяющего модуля;
- обработку результатов выполненных проверок.

Для реализации подобных процедур в неоднородных системах приведенного типа построим проверяющие и проверяемые подсистемы следующим образом. В проверяющую подсистему $U_{i,j}$ включим устройство пассивного типа (память u_i), устройство активного типа (процессор m_j) и линию связи l_{i-j} между ними, а в проверяемую подсистему $U_{j,h,k}$ включим память u_h , процессор m_k , линию связи l_{h-k} между ними, а также линию связи l_{j-h} , соединяющую процессор m_j (из подсистемы $U_{i,j}$) и память u_h . (Отметим, что номера (i, j, h, k) , записанные в индексах подсистем $U_{i,j}$, $U_{j,h,k}$, определяют номера i, h памяти и j, k процессоров, входящих в эти подсистемы.) При объединении различных устройств неоднородной системы $Q_{14, 21}$ (см. рис. 1) в подсистемы приведенного состава используем следующий прием: каждую память u_i ($i = 0, 1, \dots, n - 1$) объединим с p процессорами m_{j_1}, \dots, m_{j_p} , непосредственно связанными с этой памятью u_i , и образуем проверяющие подсистемы $U_{i,j_1}, \dots, U_{i,j_p}$. В результате таких объединений для рассматриваемой системы получим np (т. е. 21) подсистем $U_{i,j}$, различающихся номерами входящих в них устройств памяти и/или процессоров. В табл. 1 для системы $Q_{14, 21}$ приведен состав семи (из 21-й возможных) проверяющих подсистем $U_{i,j}$ (M_j) и 14-ти (из 42-х возможных) проверяемых ими подсистем U_{j,h_1,k_1} (M_{j,k_1}), U_{j,h_2,k_2} (M_{j,k_2}); (в скобках показаны более компактные символы подсистем, используемые в примере).

Проверка подсистемы $U_{j,h,k}$ (с помощью теста $t_{j,h,k}$) выполняется следующим образом. Получив управление (в процессе контроля или СД), процессор m_j подсистемы $U_{i,j}$ передает по линии l_{j-h} в память u_h команду на запуск процесса самоконтроля в подсистеме $U_{j,h,k}$. Процессор m_k , выбрав из памяти u_h и расшифровав эту команду, проводит самоконтроль подсистемы $U_{j,h,k}$ с помощью тестовых программ, хранящихся в памяти этой подсистемы. (Естественно, что при этом не проверяется линия l_{j-h} , соединяющая подсистемы $U_{i,j}$ и $U_{j,h,k}$, но ее состояние влияет на информацию, передаваемую процессором m_j в память u_h и принимаемую из нее). Резуль-



Таблица T_s состава проверяющих и проверяемых подсистем

| Строка $T_{i,j}(T_j)$ | Проверяющая подсистема | Проверяемые подсистемы и оценки их проверок | | | |
|-----------------------|---------------------------|---|-------------------------|---------------------------------------|-------------------------|
| | $U_{i,j}(M_j)$ | $U_{j,h1,k1}(M_{j,k1})$ | $r_{j,h1,k1}(r_{j,k1})$ | $U_{j,h2,k2}(M_{j,k2})$ | $r_{j,h2,k2}(r_{j,k2})$ |
| $T_{1,2}(T_0)$ | 1, (1–2), 2 (M_0) | (2–3), 3, (3–4), 4 ($M_{0,1}$) | | (2–7), 7, (7–8), 8 ($M_{0,3}$) | |
| $T_{3,4}(T_1)$ | 3, (3–4), 4 (M_1) | (4–5), 5, (5–6), 6 ($M_{1,2}$) | | (4–9), 9, (9–10), 10 ($M_{1,4}$) | |
| $T_{5,6}(T_2)$ | 5, (5–6), 6 (M_2) | (6–7), 7, (7–8), 8 ($M_{2,3}$) | | (6–11), 11, (11–12), 12 ($M_{2,5}$) | |
| $T_{7,8}(T_3)$ | 7, (7–8), 8 (M_3) | (8–9), 9, (9–10), 10 ($M_{3,4}$) | | (8–13), 13, (13–0), 0 ($M_{3,6}$) | |
| $T_{9,10}(T_4)$ | 9, (9–10), 10 (M_4) | (10–11), 11, (11–12), 12 ($M_{4,5}$) | | (10–1), 1, (1–2), 2 ($M_{4,0}$) | |
| $T_{11,12}(T_5)$ | 11, (11–12), 12 (M_5) | (12–13), 13, (13–0), 0 ($M_{5,6}$) | | (12–3), 3, (3–4), 4 ($M_{5,1}$) | |
| $T_{13,0}(T_6)$ | 13, (13–0), 0 (M_6) | (0–1), 1, (1–2), 2 ($M_{6,0}$) | | (0–5), 5, (5–6), 6 ($M_{6,2}$) | |

таты выполненного теста $t_{j,h,k}$ процессор m_k записывает в ячейки памяти u_h , откуда их, по истечению определенного времени, считывает процессор m_j , сравнивает с эталонами из собственной памяти u_i и формирует оценку $r_{j,h,k}$: 0, если результаты совпали с эталонами, 1 — при несовпадении, в том числе, из-за искажений, внесенных линией l_{j-h} . В дальнейшем эти действия кратко будем называть так: подсистема $U_{i,j}$ проверяет подсистему $U_{j,h,k}$ и формирует оценку $r_{j,h,k}$ результатов тестирования. Выполнив тесты для всех сформированных подсистем, получим реальный синдром R , объединяющий оценки проведенных тестов.

Таким образом, предлагаемый способ СД неоднородных систем основан на объединении разнотипных устройств исходной системы в подсистемы с функциональными возможностями, достаточными для выполнения процедур самоконтроля и других диагностических процедур. Далее описана реализация этого способа, основанная на методе СД для однородных систем из работ [4, 5].

3. ОРГАНИЗАЦИЯ САМОДИАГНОСТИРОВАНИЯ КОМПОНЕНТОВ НЕОДНОРОДНЫХ ЦИФРОВЫХ СИСТЕМ

Контроль работоспособности однородной цифровой системы заключается в проверке каждым проверяющим модулем технического состояния своих приемников и соответствующих линий связи. Простейший способ выполнения проверок одного модуля разными проверяющими модулями — последовательный. Для его оптимальной реализации на диагностическом графе однородной системы нужно построить гамильтонов цикл, который обеспечивает однократный обход всех модулей исправной системы. В неоднородных системах функции проверяющих модулей исполняют проверяющие подсистемы $U_{i,j}$, включающие в себя активные устройства m_j (процессоры в системе $Q_{14,21}$). Поэтому в неоднородных системах нужно строить цикл, проходящий через все проверяющие подсистемы.

Исходную информацию о составе выбранных проверяющих и проверяемых подсистем представим в виде

таблицы состава T_s , в каждой строке $T_{i,j}$ которой записан состав проверяющей подсистемы $U_{i,j}$ проверяемых ею подсистем $U_{j,h1,k1}$, $U_{j,h2,k2}$, ...; также есть поля для записи оценок проверок этих подсистем $r_{j,h1,k1}$, $r_{j,h2,k2}$, ...

Пример такой таблицы, построенной для системы $Q_{14,21}$ (см. рис. 1), показан в табл. 1. Номера i, j, h, k , записанные в клетках этой таблицы, определяют номера памяти u_i , u_h и процессоров m_j , m_k , участвующих в реализации теста $t_{j,h,k}$, а пары чисел в скобках определяют номера линий, связывающих эти памяти и процессоры. Подчеркнем, что структура графа $Q_{14,21}$ позволяет для каждой проверяющей подсистемы построить четыре различных проверяемых подсистемы. Однако в каждой строке табл. 1 показаны только две подсистемы, в составе которых и проверяются линии связи, смежные с процессором каждой проверяющей подсистемы.

Для отображения местонахождения проверяющих подсистем в цикле контроля будем записывать в строке $T_{i,j}$ подсистемы в таком порядке, что именно проверяющая подсистема $U_{h1,k1}$ (“ядро” проверяемой подсистемы $U_{j,h1,k1}$) является в построенном цикле соседней справа для проверяющей подсистемы $U_{i,j}$. Например, в строке $T_{3,4}$ табл. 1 подсистема $U_{3,4}$ проверяет подсистемы $U_{4,5,6}$ и $U_{4,9,10}$, и в следующей строке $T_{5,6}$ в качестве проверяющей записана подсистема $U_{5,6}$ (“ядро” проверяемой подсистемы $U_{4,5,6}$), являющаяся в цикле контроля соседом справа для подсистемы $U_{3,4}$.

Традиционно процесс контроля запускается диагностическим монитором (ДМ) через сравнительно небольшие интервалы времени, что препятствует накоплению отказавших компонентов в системе.

Назначим в качестве начальной подсистемы процесса контроля одну из подсистем, обозначим ее $U_{i,n}$. Для минимизации числа повторных пусков системы контроля с участием человека-оператора целесообразно назначить в качестве начальной ту подсистему, в которой произведена замена отказавшего компонента, обнаруженного в последнем цикле СД. Этот компонент имеет

наименьшее время работы в системе и, следовательно, наименьшую вероятность отказа.

По сигналу ДМ о начале процесса контроля подсистема $U_{i,n}$ выполняет следующие действия:

- из таблицы T_S выделяет строку $T_{i,n}$, находит в ней индексы $(n, h1, k1)$, $(n, h2, k2)$, ... проверяемых подсистем $U_{n,h1,k1}$, $U_{n,h2,k2}$, ...; проверяет эти подсистемы и формирует оценки $r_{n,h1,k1}$, $r_{n,h2,k2}$, ..., равные 0, если результаты тестов совпали с эталонами, и равные 1 — при несовпадении, в том числе, из-за искажений, внесенных линиями связи $U_{i,n}$ с подсистемами $U_{n,h1,k1}$, $U_{n,h2,k2}$, ...;
- сформированные оценки записывает в соответствующие клетки строки $T_{i,n}$, анализирует их:
 - если все оценки — нулевые ($r_{n,h1,k1} = 0$, $r_{n,h2,k2} = 0$, ...), то через память u_{h1} передает подсистеме $U_{h1,k1}$ (“ядру” подсистемы $U_{n,h1,k1}$) управление продолжением контроля;
 - если среди полученных оценок есть как единичные, так и нулевые (например, $r_{n,h2,k2} = 0$), то передает одной из подсистем с нулевой оценкой (например, подсистеме $U_{h2,k2}$) управление для продолжения процесса контроля;
 - если тесты самоконтроля всех проверявшихся подсистем дали единичные оценки, то ДМ запускает процесс контроля с другой проверяющей подсистемы (например, с подсистемы $U_{i,n2}$, находящейся в “середине” выбранного цикла контроля); ДМ запускает процесс контроля с подсистемы $U_{i,n2}$ и в том случае, если из-за неисправности начальной подсистемы $U_{i,n}$ процесс контроля остановился и к заданному сроку ДМ не получил сигнала об окончании СД.

Подсистема $U_{h1,k1}$ ($U_{h2,k2}$) и каждая из последующих подсистем — “соседей” по циклу контроля — выполняют аналогичные действия в случае получения нулевых оценок. Этот процесс проверок и “хождения” по вершинам выбранного цикла заканчивается, когда управление возвращается к начальной подсистеме $U_{i,n}$ ($U_{i,n2}$). Если же на одном из этапов контроля (кроме начального) получена хотя бы одна единичная оценка, свидетельствующая об обнаружении одного или нескольких отказавших компонентов, то процесс контроля прерывается, и система переходит к процессу СД отказавших компонентов, который включает в себя:

- исполнение исправной подсистемой функций проверяющей;
- нахождение пути передачи управления и информации от одной исправной проверяющей подсистемы к другой;
- обработку результатов выполненных тестов.

На основе информации, полученной в процессе контроля, ДМ запускает процесс СД с той подсистемы $U_{i,j}$, которая первой получила хотя бы одну единичную оценку при проверке состояний связанных с ней подсистем $U_{j,h1,k1}$, $U_{j,h2,k2}$, ...; естественно, подсистема $U_{i,j}$ должна быть исправной и проверенной, в частности,

она не может быть начальной подсистемой процесса контроля $U_{i,n}$ ($U_{i,n2}$).

Начальная подсистема $U_{i,j}$ ($j \neq n, n2$) выполняет следующие действия:

- формирует промежуточную таблицу $PT_{i,j}$, переписывая в нее строку $T_{i,j}$ с оценками, полученными в процессе контроля (в индексе символа данной таблицы указан индекс проверяющей подсистемы $U_{i,j}$);
- по нулевым оценкам в строке $T_{i,j}$ выделяет подмножество индексов исправных подсистем $\{q_{i,j}\}$ и одной из этих подсистем (первой в строке $T_{i,j}$, если она исправна) передает таблицу $PT_{i,j}$ и управление процессом СД.

Если среди подсистем $U_{j,h1,k1}$, $U_{j,h2,k2}$, ..., проверенных подсистемой $U_{i,j}$, нет ни одной исправной, то процесс СД не может быть продолжен, и для получения диагностических результатов ДМ должен повторить запуск процесса контроля с другой начальной подсистемы (например, с подсистемы $U_{i,n2}$).

Очередная исправная подсистема $U_{i,v}$, получив управление процессом СД, выполняет следующие действия:

- 1) из таблицы состава T_S выделяет строку $T_{i,v}$, переписывает ее в таблицу $PT_{i,j}$, полученную от предшественника — подсистемы $U_{i,j}$ ($U_{i,v-1}$);
- 2) в строке $T_{i,v}$ выделяет индексы проверяемых подсистем $U_{v,h1,k1}$, $U_{v,h2,k2}$, ..., проверяет эти подсистемы и в клетки $r_{v,h1,k1}$, $r_{v,h2,k2}$, ... строки $T_{i,v}$ записывает оценки результатов выполненных тестов;
- 3) по нулевым оценкам проверок в строке $T_{i,v}$ выделяет подмножество $\{q_{i,v}\}$ индексов исправных “преемников” — проверенных подсистем, сравнивает индексы из подмножества $\{q_{i,v}\}$ с индексами строк в таблице $PT_{i,j}$:
 - если один индекс (или больше) из подмножества $\{q_{i,v}\}$ отсутствует в списке индексов строк в таблице $PT_{i,j}$, то передает подсистеме с этим индексом таблицу $PT_{i,j}$ и управление СД;
 - если список $\{q_{i,v}\}$ пуст или все индексы $\{q_{i,v}\}$ — в списке индексов строк в таблице $PT_{i,j}$, то переход к п. 4;
- 4) находит одну из исправных подсистем $U_{i,v+1}$, $U_{i,v+2}$, ..., $U_{i,v+w}$, еще не исполнявших функции проверяющей подсистемы, и передает ей таблицу $PT_{i,j}$ и управление СД; если при выполнении этой процедуры выяснится, что все обнаруженные исправные подсистемы уже исполняли функции проверяющих, то переход к п. 5;
- 5) передает информацию, собранную в таблице $PT_{i,j}$ подсистемы $U_{i,v+w}$, в направлении, обратном тому, по которому эта информация собиралась; получив идентичную информацию в виде таблицы $PT_{i,j}$ от подсистемы $U_{i,v+w}$, каждая из исправных подсистем выполняет ее обработку и формирует одинаковый диагноз.

Для нахождения одной из исправных подсистем $U_{i,v+1}$, $U_{i,v+2}$, ..., $U_{i,v+w}$, еще не исполнявших функций



проверяющей подсистемы, подсистема $U_{i,v}$ находит индексы $\{q_{i,v^*}\}$ исправных подсистем строки T_{i,v^*} и сравнивает их с индексами строк в таблице $PT_{i,j}$. Если при сравнении списков индексов окажется, что:

- один индекс (или больше) из подмножества $\{q_{i,v^*}\}$ отсутствует в списке индексов строк в таблице $PT_{i,j}$, то подсистема $U_{i,v}$ передает подсистеме с этим индексом $(i, v + 1)$ таблицу $PT_{i,j}$ и управление СД;
- список $\{q_{i,v^*}\}$ пуст или все индексы $\{q_{i,v^*}\}$ — в списке индексов строк в таблице $PT_{i,j}$, то переход к анализу предыдущей строки T_{i,v^*-1} .

Эти действия проводятся поочередно снизу вверх, начиная с нижней строки $T_{i,v}$. При нахождении исправной подсистемы $U_{i,v+1}$ в строке T_{i,v^*} управление СД и таблица $PT_{i,j}$ передаются этой подсистеме снизу вверх через проверяющие подсистемы $U_{i,v^*}, \dots, U_{i,v}$. Поиск прекращается, если при достижении верхней строки $T_{i,j}$ окажется, что и в этой строке нет исправной подсистемы, не исполнявшей функций проверяющей подсистемы.

Для обработки информации, собранной в таблице $PT_{i,j}$ подсистемы $U_{i,v+w}$, каждая исправная подсистема переписывает ее в диагностическую таблицу T_D следующего вида. В клетках левого столбца таблицы T_D перечислены компоненты проверяемых подсистем $U_{j,h,k} (M_{j,k})$, взятые из таблицы состава T_S , а клетки следующего столбца предназначены для записи оценок $r_{j,h,k} (r_{j,k})$ проверки этих подсистем, полученных в процессе СД. Последующие столбцы таблицы T_D обозначены номерами компонентов системы, с точностью до которых диагностируются подсистемы M_k и линии L_{j-k} между подсистемами M_j и M_k . Вхождение компонентов M_k, L_{j-k} в состав подсистемы M_j , приведенной в левом столбце таблицы T_D , отмечается знаком *, поставленным в клетке на пересечении соответствующего столбца и строки $M_{j,k}$. В нижнюю часть таблицы T_D добавлены три строки, озаглавленные (в левом столбце) символами A_0, A_1, A_z и предназначенные для записи результатов обработки полученной информации. Пример такой таблицы, построенной для системы $Q_{14,21}$, показан в табл. 2, в

Таблица 2

Диагностическая таблица T_D для Примера

| Проверяемые подсистемы $M_{j,k}$ | $r_{j,k}$ | Компоненты системы | | | | | | | | | | | | | | | | | | | | | |
|----------------------------------|-----------|--------------------|--------|-------|--------|-------|--------|-------|--------|-------|--------|-------|--------|-------|--------|--------|--------|--------|--------|--------|--------|--------|---|
| | | M_0 | $L0-1$ | M_1 | $L1-2$ | M_2 | $L2-3$ | M_3 | $L3-4$ | M_4 | $L4-5$ | M_5 | $L5-6$ | M_6 | $L6-0$ | $L0-3$ | $L1-4$ | $L2-5$ | $L3-6$ | $L4-0$ | $L5-1$ | $L6-2$ | |
| $M_{0,1}$ | 0 | | 0 | 0 | | | | | | | | | | | | | | | | | | | |
| $M_{0,3}$ | 0 | | | | | | 0 | | | | | | | | 0 | | | | | | | | |
| $M_{1,2}$ | 0 | | | | 0 | 0 | | | | | | | | | | | | | | | | | |
| $M_{1,4}$ | 1 | | | | | | | | 1 | | | | | | | | 1 | | | | | | |
| $M_{2,3}$ | 0 | | | | | 0 | 0 | | | | | | | | | | | | | | | | |
| $M_{2,5}$ | 0 | | | | | | | | | | 0 | | | | | | 0 | | | | | | |
| $M_{3,4}$ | 0 | | | | | | | 0 | 0 | | | | | | | | | | | | | | |
| $M_{3,6}$ | 1 | | | | | | | | | | | | 1 | | | | | 1 | | | | | |
| $M_{4,5}$ | 0 | | | | | | | | | 0 | 0 | | | | | | | | | | | | |
| $M_{4,0}$ | 0 | 0 | | | | | | | | | | | | | | | | | | 0 | | | |
| $M_{5,6}$ | 1 | | | | | | | | | | | 1 | 1 | | | | | | | | | | |
| $M_{5,1}$ | 0 | | | 0 | | | | | | | | | | | | | | | | | 0 | | |
| $M_{6,0}$ | — | * | | | | | | | | | | | | | * | | | | | | | | |
| $M_{6,2}$ | — | | | | | * | | | | | | | | | | | | | | | | | * |
| A_0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | 0 | | 0 | | 0 | 0 | | | |
| A_1 | | | | | | | | | | | | | 1 | | | | 1 | | | | | | |
| A_z | | | | | | | | | | | | Z | | Z | | | | Z | | | | Z | |

клетках второго столбца которой записаны оценки проверок для приводимого ниже Примера. Правила обработки записей таблицы T_D и интерпретации полученных результатов даны ниже.

Процедура обработки информации из таблицы $PT_{i,j}$, выполняемая каждой исправной подсистемой, состоит из следующих шагов:

- из строк $T_{i,j}, \dots, T_{i,v+w}$ таблицы $PT_{i,j}$ выделяются оценки проверок $r_{j,h1,k1}, r_{j,h2,k2}, \dots, r_{v+w,h1,k1}$ и записываются в соответствующие клетки столбца $r_{j,h,k}$ таблицы T_D , оставшиеся клетки этого столбца заполняются прочерками (—);
- “размножаются” 0 и 1, записанные в столбце $r_{j,h,k}$ путем записи идентичных значений (0 или 1) в клетки той же строки $T_{i,j}$, отмеченные знаком *;
- заполняются прочерками (—) клетки строк A_0, A_1, A_Z ;
- выполняется следующая обработка записей таблицы T_D :
 - в клетку на пересечении строки A_0 и k -го столбца записывается 0, если в вышележащих клетках этого столбца есть хотя бы один нуль;
 - в клетку на пересечении строки A_1 и столбца, озаглавленного символом одной из вершин графа (памяти u_h , процессора m_k , подсистемы M_k), записывается 1, если в вышележащих клетках этого столбца есть хотя бы одна единица и нет ни одного нуля;
 - обрабатываются записи оставшихся столбцов, озаглавленных символами линий связи l_{j-h}, l_{h-k} (l_{j-k}) и содержащих только единицы, следующим образом:
 - если обе вершины, которые связывает рассматриваемая линия, исправны (в соответствующих клетках строки A_0 записаны 0), то в клетку на пересечении строки A_1 и данного столбца записывается 1;
 - если хотя бы одна из вершин, которые связывает рассматриваемая линия, подозревается в отказе (в соответствующей клетке строки A_1 записана 1), то в клетку на пересечении строки A_Z и данного столбца записывается Z;
- расшифровываются записи строк A_0, A_1 и A_Z :
 - в соответствии с местоположением нулевых записей в строке A_0 составляется список исправных компонентов (активных и пассивных устройств, линий связи);
 - в соответствии с местоположением единиц в строке A_1 составляется список вершин (активных и пассивных устройств) и дуг (линий связи), подозреваемых в отказе;
 - в соответствии с местоположением символов Z в строке A_Z составляется список дуг (линий связи), состоянием которых маскируется неисправным состоянием хотя бы одной из вершин, которые она соединяет;
 - прочерки (—), оставшиеся в клетках на пересечении строк A_0, A_1 и A_Z с k -м столбцом, означают, что полученной информации недостаточно для оп-

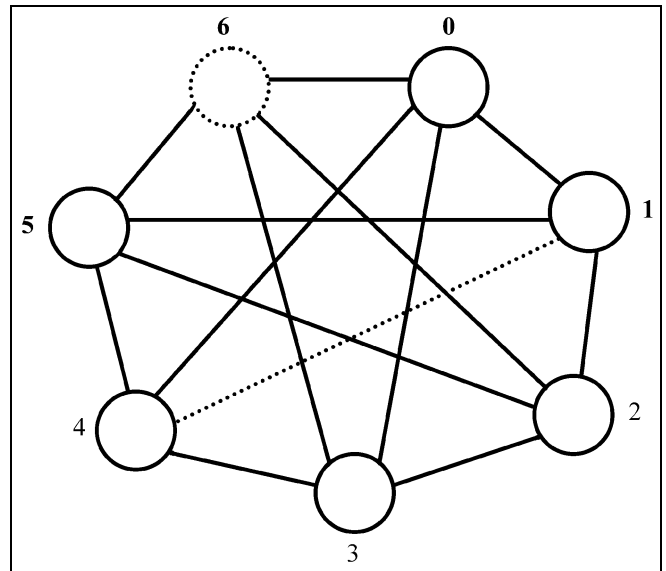


Рис. 2. Граф однородной системы $Q_{7,14}$

ределения технического состояния данного компонента (вершины, дуги графа). ♦

Предлагаемая организация СД обеспечивает получение диагностических заключений при выполнении следующих требований к составу исправных компонентов:

- в системе должно быть не менее двух исправных проверяющих подсистем и исправная линия связи между ними;
- у каждой исправной подсистемы должна быть исправна хотя бы одна из p ее входных линий.

Так как восстановление работоспособности компонентов, подозреваемых в отказе, осуществляется после выполнения СД, то данную организацию СД следует отнести к методам однократного диагностирования. Поэтому для отказовых ситуаций, которым соответствуют неразличимые потенциальные синдромы, можно получить неполный, а иногда и неправильный диагноз.

ПРИМЕР

Особенности разработанной организации СД покажем на примере неоднородной системы [2] с графом $Q_{14,21}$ (см. рис. 1). В работе [2] автор использовал неоднородную систему (см. рис. 1) для иллюстрации возможностей разработанного им алгоритма построения отказоустойчивых систем с реконfigurацией. Для этого он объединил две разнотипные вершины (память и процессор) исходного графа в одну вершину, названную агрегированной; в результате такого объединения граф неоднородной системы превратился в граф однородной системы $Q_{7,14}$ (рис. 2). Обозначим объединенную (агрегированную) подсистему символом M_j , если она выполняет функции проверяющей подсистемы, и символом M_k при вхождении ее в проверяемую подсистему M_j, k , а линию связи между подсистемами M_j и M_k символом l_{j-k} ($j \neq k = 0, \dots, 6$). В работе [2] выбран вариант, при



котором в подсистему M_j (M_k) объединены смежные процессор и память, лежащие на выбранном гамильтоновом пути. Состав полученных модулей показан в столбце M_j табл. 1. При построении отказоустойчивых систем единицей резервирования считается агрегированная подсистема, поэтому естественно считать, что разрешающая способность диагностирования отказавших компонентов должна быть равна также одному заменяемому компоненту (подсистеме или линии связи) [2, 5]. Как отмечалось, в табл. 1 приведена таблица состава T_S , предназначенная для организации тестирования в системе $Q_7, 14$.

Для проведения контроля выберем цикл, охватывающий семь подсистем M_0, \dots, M_6 в порядке возрастания их номеров, и при исправном состоянии всех компонентов управление процессом контроля от подсистемы M_6 вернется к подсистеме M_0 .

Предположим, что за время между сеансами контроля в системе накопились отказы подсистемы M_6 и линии связи (1–4) (эти компоненты показаны пунктиром на рис. 2). По сигналу ДМ о начале контроля подсистема M_0 выполнит следующие действия: из таблицы T_S (см. табл. 1) выделит строку T_0 , найдет в ней индексы (0, 1) и (0, 3) проверяемых подсистем $M_{0,1}$ и $M_{0,3}$, проверит их и сформирует оценки $r_{0,1} = r_{0,3} = 0$, которые запишет в соответствующие клетки строки T_0 (табл. 3); так как обе оценки нулевые, то передаст подсистеме M_1 (“ядру” проверенной исправной подсистемы $M_{0,1}$) управление продолжением контроля (в строке T_0 полужирным шрифтом выделен 0, по которому передается управление; таким обозначением будем пользоваться и в дальнейшем).

Подсистема M_1 , действуя аналогично, проверит подсистемы $M_{1,2}$ и $M_{1,4}$, получит нулевую и единичную оценки (показаны в строке T_1 табл. 3) и перейдет к процессу СД (для сокращения числа таблиц будем записывать оценки, находящиеся в разных подсистемах, в общей табл. 3).

В начале процесса СД подсистема M_1 выполнит следующие действия:

- сформирует промежуточную таблицу PT_1 (в табл. 3), в которую переписет строку T_1 с оценками, полученными в процессе контроля;
- по нулевой оценке ($r_{1,2} = 0$) выделит номер 2 исправной подсистемы M_2 , которой и передаст таблицу PT_1 и управление продолжением процесса СД.

Подсистема M_2 , выполнит следующие действия:

- из таблицы состава T_S (см. табл. 1) выделит строку T_2 , переписет ее в таблицу PT_1 ;
- в строке T_2 выделит индексы (2, 3) и (2, 5) проверяемых подсистем $M_{2,3}$ и $M_{2,5}$, проверит их и в соответствующие клетки строки T_2 запишет оценки $r_{2,3} = r_{2,5} = 0$;
- по нулевым оценкам выделит подмножество $\{q_2 = 3, 5\}$ номеров исправных “преемников”, сравнит эти номера с индексами строк 1, 2 в таблице

PT_1 : так как эти числа не совпадают, то передаст подсистеме M_3 таблицу PT_1 и управление СД.

Подсистема M_3 , а затем и каждая из подсистем M_4 и M_5 проверит своих преемников, и на основании полученных оценок передаст таблицу PT_1 и управление одному из исправных преемников (результаты этих действий показаны в соответствующих строках табл. 3). В отличие от предшественников, подсистема M_5 не имеет среди своих преемников исправной подсистемы, не исполнявшей функций проверяющей. Но такая подсистема M_0 , входящая в состав подсистемы $M_{4,0}$ есть в строке T_4 таблицы PT_1 . Поэтому подсистема M_5 передаст таблицу PT_1 и управление СД подсистеме M_0 через подсистему M_4 . Проверив своих преемников ($M_{0,1}$ и $M_{0,3}$), подсистема M_0 не найдет больше исправных подсистем, не исполнявших функций проверяющих. Следовательно, сбор тестовой информации закончен, и по цепочке: $M_0 \rightarrow M_5 \rightarrow M_4 \rightarrow M_3 \rightarrow M_2 \rightarrow M_1$ подсистема M_0 отправит таблицу PT_1 , чтобы каждый из исправных процессоров получил одинаковую информацию.

Для обработки информации из таблицы PT_1 исправный процессор переписывает ее в диагностическую таблицу T_D (см. табл. 2).

Обработка записей из таблицы PT_1 , выполняемая каждым исправным процессором, заключается в следующем:

- из строк T_1, \dots, T_0 таблицы PT_1 выделяются оценки проверок $r_{1,2} = 0, r_{1,4} = 1, \dots, r_{0,1} = 0, r_{0,3} = 0$, и записываются в соответствующие клетки столбца $r_{j,k}$ таблицы T_D (см. табл. 2);
- значения 0 и 1, записанные в столбце $r_{j,k}$, переписываются в клетки строки $M_{j,k}$, отмеченные знаком “*”

Таблица 3

Информация к Примеру

| Строка T_j | Проверяющая подсистема | Проверяемые подсистемы оценки их проверок | | | |
|--------------------------------|------------------------|---|-------------|-------------|-------------|
| | M_j | $M_{j, k1}$ | $r_{j, k1}$ | $M_{j, k2}$ | $r_{j, k2}$ |
| Строки с результатами контроля | | | | | |
| T_0 | M_0 | $M_{0,1}$ | 0 | $M_{0,3}$ | 0 |
| T_1 | M_1 | $M_{1,2}$ | 0 | $M_{1,4}$ | 1 |
| Таблица PT_1 | | | | | |
| T_1 | M_1 | $M_{1,2}$ | 0 | $M_{1,4}$ | 1 |
| T_2 | M_2 | $M_{2,3}$ | 0 | $M_{2,5}$ | 0 |
| T_3 | M_3 | $M_{3,4}$ | 0 | $M_{3,6}$ | 1 |
| T_4 | M_4 | $M_{4,5}$ | 0 | $M_{4,0}$ | 0 |
| T_5 | M_5 | $M_{5,6}$ | 1 | $M_{5,1}$ | 0 |
| T_0 | M_0 | $M_{0,1}$ | 0 | $M_{0,3}$ | 0 |
| Конец СД | | | | | |

(так как в данном сеансе СД подсистемы $M_{6,0}$ и $M_{6,2}$ не проверялись, то в клетках на пересечении этих строк и столбца $r_{j,k}$ остались прочерки, а в соответствующих клетках других столбцов — знаки “*”);

- заполняются прочерками (—) клетки строк A_0, A_1, A_2 ;
- результаты диагностирования формируются следующим образом:
 - в k -ю клетку строки A_0 записывается 0, если в вышележащих клетках этого столбца есть хотя бы один нуль;
 - в клетку на пересечении строки A_1 и столбца, озаглавленного символом вершины M_6 , записывается 1, так как в вышележащих клетках этого столбца есть одна единица и нет ни одного нуля;
 - в клетку на пересечении строки A_1 и столбца, озаглавленного символом линии $L(1-4)$, записывается 1, так как вершины M_1 и M_4 , которые связывает эта линия, исправны (в соответствующих клетках строки A_0 записаны нули);
 - в четыре клетки строки A_2 ; озаглавленных символами дуг $L(j-k)$, записывается символ Z , так как одна из вершин (M_6), которые связывает каждая из этих линий, подозревается в отказе (в соответствующей клетке строки A_1 записана 1);
- расшифровка записей этих строк показывает, что:
 - исправны 15 компонентов, номера которых определяются расположением нулевых записей в строке A_0 ;
 - в соответствии с расположением единиц в строке A_1 подозревается в отказе подсистема M_6 и линия связи (1—4), что совпадает с номерами компонентов в исходной отказовой ситуации;
 - в соответствии с расположением символов Z в строке A_2 остались неопределенными состояния линий (5—6), (6—0), (3—6) и (6—2), так как их проявления “маскируются” неисправным состоянием вершины 6, с которой они связаны.

Таким образом, в данном примере идентифицированы состояния 17-ти из 21-го компонента неоднородной системы $Q_{7,14}$.

ЗАКЛЮЧЕНИЕ

Представлен способ СД компонентов неоднородных цифровых систем, содержащих равное число устройств двух типов (процессоров, памяти). Для проведения процессов контроля и СД связанные разнотипные устройства объединяются в подсистемы: проверяющие и проверяемые. В основу организации СД, реализующего предложенный способ, как и в работе [4], положен принцип постепенного расширения множества исправных модулей: выполнение необходимых процедур начинается с исправной подсистемы и последовательно передается другим подсистемам, исправность которых подтверждена предшествующими проверками. Отметим, что начальная подсистема процесса СД не является

фиксированной, а определяется по результатам работы системы контроля.

Предполагается, что в исходной неоднородной системе есть диагностический монитор, который инициирует процессы контроля и СД.

В рассмотренном примере организация контроля и СД отказавших компонентов в неоднородной 14-модульной системе осуществляется с точностью до подсистемы, включающей в себя процессор, память и линию связи между ними, или линии, связывающей разные подсистемы. Требование такой точности целесообразно при диагностировании отказоустойчивых систем с реконфигурацией, в которых отказавшая подсистема автоматически заменяется исправной резервной [2, 5]. Автором был рассмотрен и другой пример организации контроля и СД отказавших компонентов в той же системе с точностью до отдельного компонента (процессора, памяти, линии связи). Для получения такой точности необходимо, чтобы проверяемые подсистемы различались номерами входящих в них устройств памяти или процессоров. Но увеличение числа различных проверяемых подсистем требует пропорционального увеличения числа необходимых проверок. В результате, снижение требуемой точности диагностирования (в приведенном примере) позволило в 3 раза уменьшить число необходимых проверок и существенно сократить трудоемкость последующих процедур. Хотя в приведенном примере показано правильное диагностирование отказов двух компонентов, но можно подобрать такие сочетания двух неисправных компонентов, диагноз которых даст неправильный результат. Причина в структуре таблицы состава, в которой каждая проверяемая подсистема тестирует по две проверяемых подсистемы, поэтому выбранные системы проверок обеспечивают достоверное диагностирование только одиночных компонентов.

ЛИТЕРАТУРА

1. Barsi F., Grandoni F., Maestrini P. A theory of diagnosability of digital systems // IEEE Trans. Comput. — 1976. — Vol. C-25, N 6. — P. 585—593.
2. Каравай М. Ф. Минимизированное вложение произвольных гамильтоновых графов в отказоустойчивый граф и реконфигурация при отказах. Ч. 1. Одно-отказоустойчивые структуры // Автоматика и телемеханика. — 2004. — № 12. — С. 159—177.
3. Пархоменко П. П. Определение технического состояния многопроцессорных вычислительных систем путем анализа графа синдромов // Там же. — 1999. — № 5. — С. 126—135.
4. Ведешенков В. А. Организация самодиагностирования технического состояния цифровых систем // Там же. — 2003. — № 11. — С. 165—182.
5. Ведешенков В. А. Самодиагностирование отказов цифровых систем с реконфигурацией // Проблемы управления. — 2003. — № 4. — С. 39—51.

☎ (095) 334-75-90

E-mail: vva@19.ipu.rssi.ru

