



ОРГАНИЗАЦИЯ ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ СО СТРУКТУРОЙ СИММЕТРИЧНОГО ДВУДОЛЬНОГО ГРАФА

В.А. Ведешенков

Отмечено, что особенность цифровых систем со структурой в виде симметричного двудольного графа состоит в использовании равного числа компонентов двух типов: процессоров и блоков памяти. Разработана организация диагностирования систем с такой структурой с точностью до отдельного компонента (процессора, памяти, линии связи). На время диагностирования разнотипные устройства объединяются в проверяемые подсистемы одинакового состава. Рассмотрен пример организации диагностирования системы, состоящей из семи процессоров и семи блоков памяти.

Ключевые слова: цифровые системы, симметричный двудольный граф, компонент, диагностирование, проверяемые подсистемы.

ВВЕДЕНИЕ

Поиск исследователями новых, более эффективных, чем известные, структур связей сложных информационных систем показал, что по ряду параметров структуры в виде симметричных двудольных графов предпочтительнее структур в виде полных графов и полнодоступных двудольных графов [1]. Симметричным двудольным графом называется граф, состоящий из двух подмножеств вершин X и Y равной мощности и ребер, у которых одна из концевых вершин принадлежит подмножеству $X(Y)$, а другая — подмножеству $Y(X)$. Пример такого графа показан на рисунке, где вершины подмножества X обозначены кружками, а подмножества Y — прямоугольниками. В числе возможных областей применения графов указанного семейства называют структуры многопроцессорных вычислительных систем (МВС), где, например, подмножество вершин X представляет совокупность процессорных элементов или вычислительных машин, а подмножество вершин Y — блоки или банки памяти.

Любой метод системного диагностирования технического состояния компонентов в цифровых системах (ЦС) предполагает, что модули, на которые разбивается исходная МВС, однородны и обладают функциональными возможностями, достаточными для реализации необходимых диагностических алгоритмов. Использование устройств

разных типов при построении систем со структурой в виде симметричного двудольного графа приводит к неоднородным системам, что не позволяет непосредственно применить системные методы диагностирования. Однако автоматическое диагностирование состояния компонентов подобных систем представляется необходимым, так как позволяет ускорить время восстановления работоспособности систем после обнаружения факта их неправильного функционирования. Кроме того, различие структур системы, представленной на рисунке, и неоднородной системы обработки радиолокационной информации не позволяет напрямую использовать метод самодиагностирования, предложенный в работе [2].

Далее представлен способ организации диагностирования состояния компонентов ЦС со структурой в виде симметричного двудольного графа. Он основан на подходе, описанном в работе [2]: на этапах контроля и диагностирования связанные разнотипные устройства объединяются в проверяемые подсистемы, обладающие необходимыми функциональными возможностями для реализации диагностических алгоритмов.

Напомним, что в однородных ЦС при выполнении процессов контроля и диагностирования 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 Барси — Грандони — Маестрини (БГМ) [3], используемой в данной работе, предполагается, что 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 [4]).

Как и в большинстве алгоритмов системного диагностирования, будем предполагать, что результаты проверок собираются и обрабатываются (дешифрируются) внешним устройством — диагностическим монитором (ДМ). По результатам дешифрации собранных результатов проверок обслуживающий персонал заменяет отказавшие компоненты и восстанавливает работоспособность МВС.

1. ПОСТАНОВКА ЗАДАЧИ

Исходная МВС со структурой в виде симметричного двудольного графа $Q_{n,s}$ представлена диагностическим графом, у которого n вершин подмножества X представляют процессоры системы, n вершин подмножества Y — блоки памяти, s — степень вершин двудольного графа; т. е. каждая вершина подмножества $X(Y)$ связана дугами с s вершинами подмножества $Y(X)$. (Для графа на рисунке $s = 3$.)

Будем считать, что допускаются устойчивые отказы ограниченного числа вершин и дуг, причем их неисправности таковы, что не влияют на работоспособность смежных модулей. Предполагается также, что в процессе диагностирования новые отказы не возникают. Для описания результатов тестирования компонентов используется модель БГМ. Запуском процессов диагностирования в различных подсистемах ЦС и обработкой полученных результатов занимается исправный ДМ, который находится в модуле, внешнем по отношению к диагностируемой ЦС.

Требуется разработать такой метод организации диагностирования ЦС со структурой в виде симметричного двудольного графа, который обеспечивает ДМ достоверной информацией для диагностирования состояния компонентов (процессоров, памяти, линий связи) системы.

2. СПОСОБ ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ СО СТРУКТУРОЙ В ВИДЕ СИММЕТРИЧНОГО ДВУДОЛЬНОГО ГРАФА

Процесс диагностирования компонентов однородных систем включает в себя такие процедуры [5]:

— исполнение исправным модулем функций проверяющего модуля;

— нахождение пути от одного исправного модуля к другому исправному для передачи ему функций проверяющего модуля;

— дешифрация результатов выполненных проверок.

Для реализации подобных процедур в неоднородных системах со структурой в виде симметричного двудольного графа поступим следующим образом.

Проверяющим модулем m_j назовем устройство активного типа (один из процессоров m_1, \dots, m_7 в системе, представленной на рисунке), а в проверяемую подсистему $U_{j,i,k}$ включим следующие компоненты: память p_i , процессор m_k , линию связи l_{i-k} между ними и линию связи l_{j-i} , соединяющую m_j с p_i . (Таким образом, индекс (j, i, k) подсистемы $U_{j,i,k}$ состоит из номера j проверяющего процессора, номера i проверяемой памяти и номера k проверяемого процессора.) Например, для системы на Рисунке в одну из проверяемых подсистем могут входить процессор m_1 в функции проверяющего модуля, память p_8 , процессор m_2 и линии связи l_{8-2} и l_{1-8} .

Наличие s устройств памяти, связанных с одним процессором цифровой системы со структурой симметричного двудольного графа ($s = 3$ для графа на рисунке), делает объединение различных устройств для формирования проверяемых подсистем неоднозначным. Выбор достаточно экономичного подмножества из числа всевозможных проверяемых подсистем является нетривиальной задачей, но здесь не рассматривается.

Тогда отдельная проверка (тест $t_{j,i,k}$) будет выполняться следующим образом. Получив управление в процессе диагностирования, проверяющий модуль (процессор) m_j передает по линии l_{j-i} в память p_i проверяемой подсистемы команды, необходимые для организации тестов самоконтроля в подсистеме $U_{j,i,k}$. Процессор m_k , получив и расшифровав эти команды, осуществляет контроль процессора m_k , линии l_{i-k} и памяти p_i с помощью тестовых программ, хранящихся в памяти p_i . Результаты выполненного теста $t_{j,i,k}$ процессор m_k записывает в ячейки памяти p_i , откуда их после очередного опроса памяти считывает процессор m_j , сравнивает с эталонами из собственной сверхоперативной памяти и формирует оценку $r_{j,i,k} = 0$, если результаты совпали с эталонами, 1 — в противном случае.



На основании полученных оценок процессор m_j выбирает исправную подсистему U_{j,i^*,k^*} , процессору m_{k^*} из состава которой передает управление и необходимую информацию для продолжения процесса диагностирования. Все исправные процессоры находятся в ждущем режиме и периодически опрашивают заранее определенные ячейки памяти. Процессор m_{k^*} , обнаружив в памяти p_{j^*} команду на выполнение процесса диагностирования, выполнит необходимые процедуры, а после их завершения передаст управление следующему процессору с помощью того же механизма (через память одной из проверенной им подсистем).

Таким образом, в предлагаемом способе диагностирования неоднородных ЦС со структурой симметричного двудольного графа предполагается объединение разнотипных устройств исходной системы в проверяемые подсистемы одинакового состава с функциональными возможностями, достаточными для выполнения диагностических процедур.

3. ОРГАНИЗАЦИЯ ДИАГНОСТИРОВАНИЯ КОМПОНЕНТОВ ЦИФРОВОЙ СИСТЕМЫ СО СТРУКТУРОЙ СИММЕТРИЧНОГО ДВУДОЛЬНОГО ГРАФА

Контроль работоспособности однородной цифровой системы заключается в том, что каждый проверяющий модуль должен проверить техническое состояние своих преемников 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 гамилътонов цикл, который обеспечит однократный об-

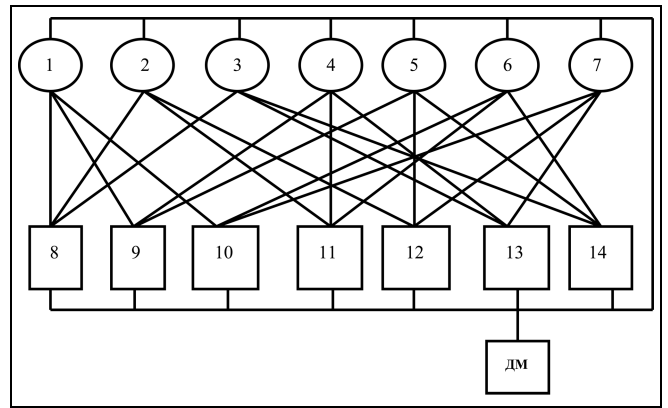


Схема цифровой системы со структурой в виде симметричного двудольного графа $Q_{7,3}$:
 1–7 — процессоры; 8–14 — блоки памяти; ДМ — диагностический монитор

ход всех модулей исправной системы. В неоднородных системах функции проверяющих модулей исполняют активные модули (процессоры m_j). Поэтому в неоднородных системах нужно построить цикл, проходящий через все активные модули (процессоры).

Но в системе со структурой в виде симметричного двудольного графа ($Q_{7,3}$ на рисунке) нельзя построить цикл, проходящий только через процессоры, так как между ними нет непосредственных связей. В цикле, выбранном для последующего примера, процессоры, номера которых выделены жирным шрифтом, чередуются с блоками памяти: $5 \rightarrow 12 \rightarrow 7 \rightarrow 10 \rightarrow 6 \rightarrow 14 \rightarrow 3 \rightarrow 13 \rightarrow 4 \rightarrow 11 \rightarrow 2 \rightarrow 8 \rightarrow 1 \rightarrow 9 \rightarrow 5$.

В таблице T_S (табл. 1) показана информация о составе проверяемых подсистем $U_{j,i,k}$ и номерах

Таблица 1

Таблица T_S состава проверяемых подсистем

Строка T_j	Проверяющий модуль	Проверяемые подсистемы $U_{j,i,k}$ оценки результатов их проверок					
	m_j	$U_{j,i1,k1}$	$r_{j,i1,k1}$	$U_{j,i2,k2}$	$r_{j,i2,k2}$	$U_{j,i3,k3}$	$r_{j,i3,k3}$
T_5	5	(5-12), 12, (12-7), 7		(5-9), 9, (9-4), 4		(5-14), 14, (14-6), 6	
T_7	7	(7-10), 10, (10-6), 6		(7-12), 12, (12-2), 2		(7-13), 13, (13-3), 3	
T_6	6	(6-14), 14, (14-3), 3		(6-11), 11, (11-2), 2		(6-10), 10, (10-1), 1	
T_3	3	(3-13), 13, (13-4), 4		(3-14), 14, (14-5), 5		(3-8), 8, (8-2), 2	
T_4	4	(4-11), 11, (11-2), 2		(4-13), 13, (13-3), 3		(4-9), 9, (9-1), 1	
T_2	2	(2-8), 8, (8-1), 1		(2-11), 11, (11-4), 4		(2-12), 12, (12-5), 5	
T_1	1	(1-9), 9, (9-5), 5		(1-8), 8, (8-2), 2		(1-10), 10, (10-7), 7	

проверяющих их модулей m_j , выбранных для последующего примера диагностирования компонентов системы $Q_{7,3}$ (см. рисунок). В таблице семь строк T_j , в каждой из которых для проверяющего модуля m_j ($j = 1, 2, \dots, 7$) записаны номера компонентов, входящих в состав подсистем $U_{j,i1,k1}$, $U_{j,i2,k2}$ и $U_{j,i3,k3}$, проверяемых модулем m_j , в ней также есть поля для записи оценок результатов проверки $r_{j,i1,k1}$, $r_{j,i2,k2}$ и $r_{j,i3,k3}$ этих подсистем модулем m_j . (Отметим, что в клетках строк T_5, \dots, T_7 табл. 1 показаны более компактные номера индексов проверяющего модуля и проверяемых подсистем.)

Для отображения местонахождения процессоров в цикле контроля в строке T_j табл. 1 проверяемые подсистемы записаны в таком порядке, что именно процессор m_{k1} , входящий в проверяемую подсистему $U_{j,i1,k1}$, является в построенном цикле соседним справа для проверяющего процессора m_j (после памяти p_{i1}). Например, в строке T_5 процессор m_5 является проверяющим для подсистем $U_{5,12,7}$, $U_{5,9,4}$ и $U_{5,14,6}$, и в следующей строке T_7 в качестве проверяющего модуля записан процессор m_7 , являющийся в цикле контроля соседом справа для процессора m_5 (после памяти p_{12}).

Поскольку задачи, возлагаемые на системы диагностирования рассматриваемых ЦС и неоднородных ЦС, рассмотренных в статье [2], во многом сходны, то в организации системы диагностирования рассматриваемых здесь ЦС можно использовать большинство диагностических процедур, описанных в статье [2]. Поэтому в последующем тексте будем опускать некоторые подробности, заменяя их необходимыми ссылками.

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

Назначим в качестве начального модуля процесса контроля один из процессоров, обозначим его m_n .

По сигналу ДМ о начале контроля модуль m_n выполняет следующие действия:

— из таблицы T_5 выделяет строку T_n , находит в ней индексы $(n, i1, k1)$, $(n, i2, k2)$, ... проверяемых подсистем $U_{n,i1,k1}$, $U_{n,i2,k2}$, ...;

— передает процессору m_{k1} (m_{k2} , ...) команду «Выполнить самоконтроль подсистемы $U_{n,i1,k1}$ ($U_{n,i2,k2}$, ...) и передать результаты проверок в модуль m_n ».

Получив эту команду, модуль m_{k1} (m_{k2} , ...) выполняет самоконтроль подсистемы $U_{n,i1,k1}$ ($U_{n,i2,k2}$, ...)

и передает полученные результаты проверок в модуль m_n .

Модуль m_n формирует оценку $r_{n,i1,k1}$ ($r_{n,i2,k2}$, ...), равную 0, если результаты тестов совпали с эталонами, и равную 1 при несовпадении; записывает ее в соответствующие клетки строки T_n , затем анализирует полученные оценки:

— если среди полученных оценок есть как единичные, так и нулевые, то передает процессору одной из подсистем с нулевой оценкой управление для продолжения процесса контроля;

— если тесты самоконтроля всех проверявшихся подсистем дали единичные оценки, то ДМ должен запустить процесс контроля с другого проверяющего модуля.

Модуль m_{k1} (m_{k2} , ...) и каждый из последующих «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 начальному модулю m_n . 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 приступает к процессу диагностирования отказавших компонентов.

На основе информации, полученной в процессе контроля, ДМ запускает процесс диагностирования с того модуля m_j , который первым получил хотя бы одну единичную оценку при проверке состояний связанных с ним подсистем $U_{j,i1,k1}$, $U_{j,i2,k2}$, ...; естественно, модуль m_j должен быть исправным и проверенным в данном «сеансе» контроля, в частности, он не может быть начальным модулем процесса контроля m_n .

Начальный модуль процесса диагностирования m_j выполняет следующие действия:

— формирует промежуточную таблицу PT_j , переписывая в нее строку T_j с оценками, полученными в процессе контроля (в индексе символа данной таблицы указан индекс проверяющего модуля m_j);

— по нулевым оценкам в строке T_j выделяет подмножество индексов исправных модулей $\{q_j\}$ и одному из этих модулей (например, первому в строке T_j , если он исправен) передает таблицу PT_j и управление процессом диагностирования.

Если среди модулей, проверенных модулем m_j , нет ни одного исправного, то для получения диагностических результатов ДМ должен повторить запуск процесса контроля с другого начального модуля.



Очередной исправный модуль m_v , получив управление процессом диагностирования, выполняет следующие действия:

1) из таблицы состава T_S выделяет строку T_v , переписывает ее в таблицу PT_j , полученную от предшественника — модуля m_j (m_{v-1});

2) в строке T_v выделяет индексы $(v, i1, k1)$, $(v, i2, k2)$... проверяемых подсистем $U_{v,i1,k1}$, $U_{v,i2,k2}$, ...; выдает команды на самоконтроль этих подсистем, получает результаты этих проверок и в клетки $r_{v,i1,k1}$, $(r_{v,i2,k2}, \dots)$, ... строки T_v записывает оценки результатов выполненных тестов;

3) по нулевым оценкам проверок в строке T_v выделяет подмножество $\{q_v\}$ индексов исправных «преемников» — проверенных модулей, сравнивает индексы из подмножества $\{q_v\}$ с индексами строк в таблице PT_j ;

— если один индекс (или больше) из подмножества $\{q_v\}$ отсутствует в списке индексов строк в таблице PT_j , то передает модулю с этим индексом таблицу PT_j и управление процессом диагностирования;

— если список $\{q_v\}$ пуст или все индексы $\{q_v\}$ — в списке индексов строк в таблице PT_j , то переход к п. 4;

4) находит один из исправных модулей, еще не исполнявших функции проверяющего, и передает ему таблицу PT_j и управление диагностированием.

Для нахождения одного из исправных модулей, еще не исполнявших функций проверяющего модуля, модуль m_v находит индексы $\{q_{v*}\}$ исправных модулей строки T_{v*} и сравнивает их с индексами строк в таблице PT_j . Если при сравнении списков индексов окажется, что

— один индекс (или больше) из подмножества $\{q_{v*}\}$ отсутствует в списке индексов строк в таблице PT_j , то модуль m_v передает модулю с этим индексом $(v + 1)$ таблицу PT_j и управление диагностированием;

— список $\{q_{v*}\}$ пуст или все индексы $\{q_{v*}\}$ — в списке индексов строк в таблице PT_j , то модуль m_v переходит к анализу предыдущей строки T_{v*-1} .

Эти действия выполняются поочередно снизу вверх, начиная с нижней строки T_v . При нахождении исправного модуля m_{v+1} в строке T_{v*} управление диагностированием и таблица PT_j передаются этому модулю снизу вверх через проверяющие модули m_v, \dots, m_{v*} . Если при достижении верхней строки T_j окажется, что и в этой строке нет исправ-

ного модуля, не исполнявшего функций проверяющего модуля, то поиск прекращается, и модуль m_v пересылает в ДМ таблицу PT_j для обработки полученных результатов проверок.

Процедура обработки полученных результатов проверок достаточно полно изложена в работе [2]. Поэтому в данном тексте эта процедура будет показана на примере.

4. ПРИМЕР

Особенности разработанной организации диагностирования покажем на примере ЦС со структурой симметричного двудольного графа отказов (система $Q_{7,3}$ на рисунке).

В цикле, выбранном для данного примера, процессоры, номера которых выделены жирным шрифтом, чередуются с блоками памяти: **5** → 12 → 7 → 10 → 6 → 14 → 3 → 13 → 4 → 11 → 2 → 8 → 1 → 9 → 5. Таблица состава T_S , предназначенная для организации тестирования в системе $Q_{7,3}$, приведена в табл. 1.

Предположим, что из-за большого интервала между циклами контроля в рассматриваемой ЦС накопились отказы двух модулей 6 и 8 и двух линий связи (4—13) и (5—12).

По сигналу ДМ о начале контроля начальный модуль 5 выполнит следующие действия: из таблицы T_S (см. табл. 1) выделит строку T_5 , найдет в ней индексы (5, 12, 7) (5, 9, 4), (5, 14, 6) проверяемых подсистем $U_{5,12,7}$, $U_{5,9,4}$, $U_{5,14,6}$, передаст модулю 7 (4, 6) команду «Выполнить самоконтроль подсистемы $U_{5,12,7}$ ($U_{5,9,4}$, $U_{5,14,6}$) и вернуть результаты проверок в модуль 5».

Получив эту команду, модуль 7 (4, 6) выполнит самоконтроль подсистемы $U_{5,12,7}$ ($U_{5,9,4}$, $U_{5,14,6}$) и передаст полученные результаты проверок модулю 5.

Модуль 5 формирует оценки $r_{5,12,7} = 1$, $r_{5,9,4} = 0$, $r_{5,14,6} = 1$, записывает их в соответствующие клетки строки T_5 (для сокращения числа таблиц будем записывать результаты проверок, находящиеся в разных модулях, в общую табл. 2), анализирует полученные оценки: так как среди полученных оценок есть две единичные и одна нулевая, то передает модулю 4 подсистемы $U_{5,9,4}$ управление для продолжения процесса контроля. (В строке T_1 и в последующих полужирным шрифтом в результатах проверки выделен 0 того исправного модуля, которому передается управление для продолжения процесса контроля или диагностирования.)

Модуль 4 выполнит аналогичные действия:

— передаст в подсистемы $U_{4,13,3}$, $U_{4,11,6}$, $U_{4,9,1}$ команду «Выполнить самоконтроль и вернуть результаты проверок в модуль 4»;

— получит оценки результатов контроля $r_{4,13,3} = 1$, $r_{4,11,6} = 1$, $r_{4,9,1} = 0$, проанализирует их: так как среди полученных оценок есть единичные ($r_{4,13,3} = 1$, $r_{4,11,6} = 1$), то заведомо исправный модуль 4 перейдет к процессу диагностирования отказавших компонентов и выполнит следующие действия:

— образует промежуточную таблицу PT_4 и переписывает в нее строку T_4 с полученными оценками результатов проверок;

— по нулевым оценкам в строке T_4 выделит индекс исправной подсистемы $U_{4,9,1}$, в котором найдет номер исправного модуля $\{q_4 = 1\}$, которому передаст таблицу PT_4 и управление процессом диагностирования.

В свою очередь, модуль 1 выполнит следующие действия:

— из таблицы T_5 (табл. 1) выделит строку T_1 , переписывает ее в таблицу PT_4 , в строке T_1 найдет индексы проверяемых подсистем $U_{1,8,2}$, $U_{1,9,5}$ и $U_{1,10,7}$, в которые передаст команду «Выполнить самоконтроль и вернуть результаты проверок в модуль 1»;

— получит оценки результатов контроля $r_{1,8,2} = 1$, $r_{1,9,5} = 0$, $r_{1,10,7} = 0$;

— по нулевым оценкам в строке T_1 выделит индексы исправных подсистем $U_{1,9,5}$ и $U_{1,10,7}$, в которых найдет номера исправных модулей $\{q_1 = 5, 7\}$ и первому из них (модулю 5) передаст таблицу PT_4 и управление процессом диагностирования.

Модуль 5 вместо проверки подсистем $U_{5,12,7}$, $U_{5,9,4}$ и $U_{5,14,6}$ выберет из строки T_5 результаты их проверок (на этапе контроля) $r_{5,12,7} = 1$, $r_{5,9,4} = 0$, $r_{5,14,6} = 1$ и определит, что исправен модуль 4. Но так как модуль 4 уже есть в числе проверяющих модулей на этапе диагностирования, то модуль 5 должен искать исправный модуль для исполнения функций проверяющего модуля среди ранее проверенных. В данном случае в предыдущей строке T_1 есть нулевые оценки проверок модулей 5 и 7. Таким образом, модуль 5 через модули 9, 1 и 10

должен передать модулю 7 таблицу PT_4 и управление дальнейшим процессом диагностирования.

Модуль 7 выполнит необходимые проверки подсистем $U_{7,10,6}$, $U_{7,12,2}$ и $U_{7,13,3}$, определит, что среди его «преемников» есть два исправных модуля: 2 и 3, и первому из них (модулю 2) передаст таблицу PT_4 и управление дальнейшим процессом диагностирования. В свою очередь, модуль 2 после проверок своих «преемников» — подсистем $U_{2,11,4}$, $U_{2,8,1}$ и $U_{2,12,5}$ выяснит, что в этих подсистемах один исправный модуль 4, который уже есть в числе проверяющих модулей на этапе диагностирования. Поэтому, как и модуль 5, модуль 2 должен искать исправный модуль для исполнения функций проверяющего модуля среди ранее проверенных. В данном случае в предыдущей строке T_7 есть нулевые оценки проверок модулей 2 и 3. Таким образом, модуль 2 через модули 12, 7 и 13 должен передать модулю 3 таблицу PT_4 и управление дальнейшим процессом диагностирования.

В свою очередь, модуль 3 после проверок своих «преемников» — подсистем $U_{3,14,5}$, $U_{3,13,4}$ и $U_{3,8,2}$ выяснит, что исправный модуль 5 уже есть в числе проверяющих модулей на этапе диагностирования. Поэтому, как и модули 5 и 2, модуль 3 будет искать исправный модуль для исполнения функций проверяющего модуля. Не обнаружив такого модуля в расположенных выше строках таблицы PT_4 , так как все исправные модули уже исполняли подобные функции, модуль 3 закончит процесс диагностирования и перешлет в ДМ таблицу PT_4 для обработки полученных результатов проверок.

Для обработки и дешифрации результатов ДМ переписывает их из таблицы PT_4 в заранее заготовленную

Таблица 2

Информация к Примеру

Строка T_j	Проверяющий модуль m_j	Проверяемые подсистемы $U_{j,i,k}$, оценки их проверок					
		$U_{j,i,k1}$	$r_{j,i,k1}$	$U_{j,i,k2}$	$r_{j,i,k2}$	$U_{j,i,k3}$	$r_{j,i,k3}$
Строки с результатами контроля							
T_5	5	(5-12), 12, (12-7), 7	1	(5-9), 9, (9-4), 4	0	(5-14), 14, (14-6), 6	1
T_4	4	(4-13), 13, (13-3), 3	1	(4-11), 11, (11-6), 6	1	(4-9), 9, (9-1), 1	0
Таблица PT_4							
T_4	4	(4-13), 13, (13-3), 3	1	(4-11), 11, (11-6), 6	1	(4-9), 9, (9-1), 1	0
T_1	1	(1-8), 8, (8-2), 2	1	(1-9), 9, (9-5), 5	0	(1-10), 10, (10-7), 7	0
T_5	5	(5-12), 12, (12-7), 7	1	(5-9), 9, (9-4), 4	0	(5-14), 14, (14-6), 6	1
T_7	7	(7-10), 10, (10-6), 6	1	(7-12), 12, (12-2), 2	0	(7-13), 13, (13-3), 3	0
T_2	2	(2-11), 11, (11-4), 4	0	(2-8), 8, (8-1), 1	1	(2-12), 12, (12-5), 5	1
T_3	3	(3-14), 14, (14-5), 5	0	(3-13), 13, (13-4), 4	1	(3-8), 8, (8-2), 2	1



диагностическую таблицу T_D [2]. В табл. 3, а и 3, б представлены две части таблицы T_D после обработки результатов из таблицы PT_4 (см. табл. 2). Поясним структуру этих таблиц. В клетках левого столбца перечислены компоненты проверяемых подсистем $U_{j,i,k}$, взятые из соответствующих клеток таблицы состава T_S (см. табл. 1), а клетки следующего столбца предназна-

чены для записи оценок $r_{j,i,k}$ проверок этих подсистем, полученных в процессе диагностирования. Последующие столбцы T_D обозначены символами одиночных компонентов системы $Q_{7,3}$: процессоры — символами u с номерами 1 — 7, блоки памяти — символами u с номерами 8 — 14, линии связи — символами l с номерами начальной и конечной вершин. Нетрудно заметить,

Таблица 3а

Диагностическая таблица T_D

Проверяемые подсистемы $U_{j,i,k}$	$r_{j,i,k}$	Компоненты системы																	
		$u5$	$l5-12$	$u12$	$l12-7$	$u7$	$l7-10$	$u10$	$l10-6$	$u6$	$u8$	$l1-10$	$l1-8$	$u2$	$l2-8$	$u11$	$l11-2$	$l11-4$	$u4$
(5-12), 12, (12-7), 7	1		1	1	1	1													
(5-9), 9, (9-4), 4	0																		0
(5-14), 14, (14-6), 6	1								1										
(7-10), 10, (10-6), 6	1						1	1	1	1									
(7-12), 12, (12-2), 2	0			0	0									0					
(7-13), 13, (13-3), 3	0																		
(6-14), 14, (14-3), 3	—																		
(6-11), 11, (11-2), 2	—													*		*	*		
(6-10), 10, (10-1), 1	—							*	*				*						
(3-13), 13, (13-4), 4	1																		1
(3-14), 14, (14-5), 5	0	0																	
(3-8), 8, (8-2), 2	1										1			1	1				
(4-11), 11, (11-2), 2	0													0		0	0	0	
(4-13), 13, (13-3), 3	1																		
(4-9), 9, (9-1), 1	0																		
(2-8), 8, (8-1), 1	1										1		1		1				
(2-11), 11, (11-4), 4	0														0	0	0	0	0
(2-12), 12, (12-5), 5	1	1	1	1															
(1-9), 9, (9-5), 5	0	0																	
(1-8), 8, (8-2), 2	1										1		1	1	1				
(1-10), 10, (10-7), 7	0					0	0	0					0						
A_0	0	—	0	0	0	0	0	—	—	—	0	—	0	—	0	0	0	0	0
A_1	—	1	—	—	—	—	—	—	1	1	—	—	—	—	—	—	—	—	—
A_Z	—	—	—	—	—	—	—	Z	—	—	—	Z	—	Z	—	—	—	—	—



что в столбце $U_{j,i,k}$ проверяемых подсистем каждая линия показана дважды: в прямом и инверсном направлении, например (5–12) и (12–5), (12–7) и (7–12) и т. д. Однако в списке отдельных компонентов каждая из линий записана один раз для сокращения ширины табл. 3, а и 3, б.

Вхождение компонента u_k (l_{j-i}) в состав подсистемы $U_{j,i,k}$, приведенной в левом столбце таблицы T_D , отме-

чается знаком «*», поставленным в клетку на пересечении столбца u_k (l_{j-i}) и строки $U_{j,i,k}$. (Так как в табл. 3, а и 3, б показаны результаты обработки информации из PT_4 , то знаки «*» остались только в тех трех строках, для которых нет оценок проверок этих подсистем.)

В нижней части таблицы T_D находятся три строки A_0 , A_1 и A_Z , предназначенные для записи результатов обработки полученной информации.

Таблица 3б

Диагностическая таблица T_D

Проверяемые подсистемы $U_{j,i,k}$	$r_{j,i,k}$	Компоненты системы																
		$u13$	$l4-13$	$u3$	$l6-11$	$l3-14$	$u14$	$l14-5$	$l3-8$	$u9$	$l9-1$	$l3-13$	$u1$	$l4-9$	$l5-9$	$l6-14$	$l2-12$	$l7-13$
(5-12), 12, (12-7), 7	1																	
(5-9), 9, (9-4), 4	0								0				0	0				
(5-14), 14, (14-6), 6	1						1	1								1		
(7-10), 10, (10-6), 6	1																	
(7-12), 12, (12-2), 2	0																0	
(7-13), 13, (13-3), 3	0	0		0								0						0
(6-14), 14, (14-3), 3	—			*		*	*								*			
(6-11), 11, (11-2), 2	—				*													
(6-10), 10, (10-1), 1	—											*						
(3-13), 13, (13-4), 4	1	1	1									1						
(3-14), 14, (14-5), 5	0					0	0	0										
(3-8), 8, (8-2), 2	1								1									
(4-11), 11, (11-2), 2	0																	
(4-13), 13, (13-3), 3	1	1	1	1								1						
(4-9), 9, (9-1), 1	0									0	0		0	0				
(2-8), 8, (8-1), 1	1												1					
(2-11), 11, (11-4), 4	0																	
(2-12), 12, (12-5), 5	1																1	
(1-9), 9, (9-5), 5	0									0	0				0			
(1-8), 8, (8-2), 2	1												1					
(1-10), 10, (10-7), 7	0																	
A_0		0	—	0	—	0	0	0	—	0	0	0	0	0	0	—	0	0
A_1		—	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
A_Z		—	—	—	Z	—	—	—	Z	—	—	—	—	—	—	Z	—	—



Обработка записей из таблицы PT_4 заключается в следующем:

- из строк T_4 , T_1 и T_3 таблицы PT_4 выделены оценки результатов проверок $r_{4,13,3} = 1$, $r_{4,11,6} = 1$, $r_{4,9,1} = 0$, ..., $r_{3,14,5} = 0$, $r_{3,13,4} = 1$, $r_{3,8,2} = 1$ и записаны в соответствующие клетки столбца $r_{j,i,k}$ табл. 3, а и 3, б, три клетки этого столбца заполнены прочерками «—»; в этих строках и сохранились знаки «*»;
- значения 0 и 1, записанные в столбце $r_{j,i,k}$, переписаны в клетки той же строки $U_{j,i,k}$, отмеченные знаком «*»;
- заполнены прочерками «—» клетки строк A_0 , A_1 , A_2 ;
- после этого в клетки этих строк записаны результаты диагностирования в соответствии со следующими правилами:
 - в k -ю клетку строки A_0 записан нуль, если в вышележащих клетках этого столбца есть хотя бы один нуль;
 - в клетки на пересечении строки A_1 и столбцов, озаглавленных символами вершин $и6$ и $и8$, записаны единицы, так как в вышележащих клетках этих столбцов есть хотя бы одна единица и нет ни одного нуля;
 - в клетки на пересечении строки A_1 и столбцов, озаглавленных символами дуг $l(5-12)$, $l(4-13)$ записаны единицы, так как в этих столбцах нет ни одного нуля и каждая из вершин, которые связывают эти линии, исправна (в соответствующих клетках строки A_0 записан нуль),
 - в шести клетках строки A_2 , озаглавленных символами дуг $l(j-i)$, записан символ Z , так как хотя бы одна из вершин (j или i), которые связывает каждая из этих линий, подозревается в отказе (в соответствующей клетке строки A_1 записана единица);
- расшифровка полученных записей в строках A_0 , A_1 и A_2 дает следующие диагностические заключения:
 - исправны 25 компонентов, номера которых определяются местоположением нулевых записей в строке A_0 ;
 - в соответствии с местоположением единиц в строке A_1 подозревается в отказе процессор $б$, блок памяти 8 , линии связи $(5-12)$ и $(4-13)$, что совпадает с номерами компонентов в исходной отказовой ситуации;
 - в соответствии с местоположением символов Z в строке A_2 остались неопределенными состояния линий $(10-б)$, $(1-8)$, $(2-8)$, $(6-11)$, $(3-8)$, $(6-14)$, так как их проявления «маскируются» неисправным состоянием одной из вершин ($б$ или 8), которые они соединяют.

Таким образом, идентифицированы состояния 29-ти из 35-ти компонентов системы $Q_{7,3}$.

ЗАКЛЮЧЕНИЕ

Особенность цифровых систем (ЦС) со структурой в виде симметричного двудольного графа состоит в наличии компонентов разных типов, на-

пример, процессоров и блоков памяти. Неоднородность компонентов ЦС подобной структуры не позволяет напрямую применять методы системного диагностирования, разработанные для однородных ЦС.

Предложен способ диагностирования компонентов ЦС со структурой в виде симметричного двудольного графа, в основе которого лежит объединение связанных разнотипных устройств на этапах контроля и диагностирования в проверяемые подсистемы одинакового состава. С учетом полученной таким образом одинаковости состава проверяемых подсистем разработана организация диагностирования отказавших компонентов в ЦС подобной структуры.

Показан пример организации диагностирования отказавших компонентов в системе, состоящей из семи процессоров и семи блоков памяти; требуемая точность диагностирования — до отдельного компонента ЦС: процессор, память и линия связи между ними. В данном примере получен правильный результат: диагноз совпадает с исходной отказовой комбинацией из четырех компонентов. Но можно подобрать такие сочетания неисправных компонентов, диагноз которых даст неточный или неправильный результат. Работы по определению меры диагностируемости (числа правильно диагностируемых отказавших компонентов, расположенных произвольным образом) для ЦС подобной структуры неизвестны.

ЛИТЕРАТУРА

1. *Каравай М.Ф., Пархоменко П.П., Подлазов В.С.* Комбинаторные методы построения двудольных однородных минимальных квазиполных графов (симметричных блок-схем) // Автоматика и телемеханика. — 2009. — № 2. — С. 153–170.
2. *Ведешенков В.А.* Способ самодиагностирования неоднородных цифровых систем // Проблемы управления. — 2005. — № 4. — С. 33–40.
3. *Barsi F., Grandoni F., Maestrini P.* A theory of diagnosability of digital systems // IEEE Trans. Comput. — 1976. — Vol. C-25, № 6. — P. 585–593.
4. *Пархоменко П.П.* Определение технического состояния многопроцессорных вычислительных систем путем анализа графа синдромов // Автоматика и телемеханика. — 1999. — № 5. — С. 126–135.
5. *Ведешенков В.А.* Организация самодиагностирования технического состояния цифровых систем // Автоматика и телемеханика. — 2003. — № 11. — С. 165–182.

Статья представлена к публикации членом редколлегии чл.-корр. РАН П.П. Пархоменко.

Ведешенков Виктор Алексеевич — д-р техн. наук, гл. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, ☎(495) 335-75-90, ✉vva@ipu.ru.