УДК 681.518.54; 004.3.001.4

ПОДХОД К ДИАГНОСТИРОВАНИЮ ВОЗНИКШЕГО ОТКАЗА В ЦИФРОВЫХ СИСТЕМАХ СО СТРУКТУРОЙ СИММЕТРИЧНОГО ДВУДОЛЬНОГО ГРАФА

В.А. Ведешенков

Представлен подход к диагностированию возникшего отказа в цифровых системах со структурой симметричного двудольного графа. Для обнаружения таких отказов и выработки правильного диагноза осуществляется дву- и более кратный запуск процесса диагностирования и сравнение результатов проверок, полученных в смежных запусках. Диагностический монитор инициирует процессы контроля и диагностирования, обрабатывает результаты проверок. Рассмотрен пример диагностирования отказавших компонентов в системе, состоящей из семи процессоров и семи блоков памяти.

Ключевые слова: цифровая система, симметричный двудольный граф, компонент, двукратный запуск, диагностирование, проверяемая подсистема.

ВВЕДЕНИЕ

Симметричным двудольным графом называется граф, состоящий из двух подмножеств вершин X и Y равной мощности n и имеющий ребра, одна из концевых вершин которых принадлежит подмножеству X (Y), а другая — подмножеству Y (X) [1]. Пример такого графа для $n = 7$ показан на рисунке, где вершины подмножества X обозначены кружками, а подмножества Y — прямоугольниками. В числе возможных областей применения графов указанного семейства называют структуры многомашинных вычислительных систем, где, например, подмножество вершин X представляет совокупность процессорных элементов или вычислительных машин, а подмножество вершин Y — блоки или банки памяти. В работе [2] представлена организация диагностирования цифровых систем (ЦС) со структурой симметричного двудольного графа, в которой выполнение необходимых процедур начинается с исправного модуля и последовательно передается другим модулям, исправность которых подтверждена предшествующими проверками. Поэтому наиболее «вредными», приводящими к ошибкам диагностирования для предложенной организации могут быть отказы модуля, признанного ранее исправным и получившего управление для выполнения диагностических процедур после возникновения в нем отказа. Следует признать, что вероятность возникновения новых отказов в процессе диагностирования ЦС со

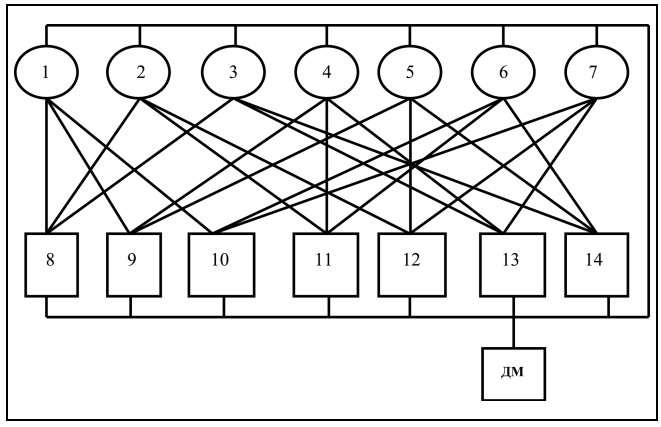

Схема цифровой системы со структурой симметричного двудольного графа $Q_{7,3}$:
1—7 — процессоры; *8—14* — блоки памяти; *ДМ* — диагностический монитор

структурой симметричного двудольного графа не слишком большая, но пренебрегать ею нежелательно. Отметим, что подобная задача самодиагностирования возникающих отказов и их устранения рассматривалась в статье [3] применительно к цифровым системам с реконфигурацией, все модули которых однотипны. Различие в структурах и модульном составе систем в работах [2, 3] не позволяет впрямую применять метод самодиагностирования, предложенный в работе [3].

Далее излагается подход к диагностированию возникшего отказа в ЦС со структурой симметричного двудольного графа, в соответствии с которым используются предложенные в работе [3] методы: для обнаружения таких отказов и выработки правильного диагноза выполняется дву- и более кратный запуск процесса диагностирования со сравнением полученных результатов проверок.

1. ПОСТАНОВКА ЗАДАЧИ

Исходная многомашинная вычислительная система со структурой 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, изображенного 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, что во время процесса диагностирования может возникнуть один новый отказ. Для описания результатов тестирования компонентов используется модель Барси—Грандони—Маестрини. Запуски процессов диагностирования в различных подсистемах ЦС и обработку полученных результатов выполняет исправный диагностический монитор (ДМ), который находится вне диагностируемой ЦС. При диагностировании разнотипные устройства исходной системы объединяются в проверяемые подсистемы одинакового состава [2]; в каждую проверяемую подсистему включаются следующие компоненты: блок памяти, процессор, линия связи между ними и линия связи, соединяющая проверяющий модуль с блоком памяти.

Требуется разработать подход к диагностированию возникшего отказа в ЦС со структурой симметричного двудольного графа, обеспечивающий ДМ достоверной информацией для диагностирования состояния компонентов (процессоров, блоков памяти, линий связи) системы.

2. ПОДХОД К ДИАГНОСТИРОВАНИЮ ВОЗНИКШЕГО ОТКАЗА

Для обнаружения отказа, возникшего в процессе диагностирования и получения правильного диагноза будем использовать двух- и более кратный запуск процесса со сравнением результатов двух соседних запусков. Совпадение результатов первого и второго запусков свидетельствует о том, что неисправности компонентов, подозреваемых в отказе, имели место до первого запуска. Если же эти результаты не совпали, то потому, что в процессе первого или второго запусков проявился отказ еще одного из компонентов системы. В таком случае ДМ должен повторять процессы диагностирования до тех пор, пока не совпадут результаты проверок в двух соседних запусках: втором и третьем, третьем и четвертом и т. д.

Для повышения достоверности получаемых результатов проверок в качестве начального модуля второго, третьего и последующих запусков процесса диагностирования будем назначать тот модуль, который был последним в цепочке проверяющих модулей предыдущего запуска, т. е. первого, второго и т. д. Таким выбором начального модуля обеспечивается максимум вероятности того, что очередной запуск будет начат с исправного проверяющего модуля.

Начальным модулем первого запуска ДМ назначает тот модуль, который был заменен на исправный при последнем восстановлении системы; этот модуль имеет наименьшую наработку на отказ и, следовательно, наименьшую вероятность отказа на момент включения системы контроля и диагностирования. Начальный модуль может назначаться и фиксированно, например, как первый модуль цикла контроля. В любом варианте, исправность назначенного модуля проверяется на первом шаге процедуры: при выполнении этим модулем функций проверяющего модуля по отношению к своим проверяемым подсистемам. Если результаты проверок этих подсистем окажутся неудовлетворительными, например, все системы признаны неисправными, то начальный модуль прекращает свою работу. Диагностический монитор, не получив от системы результатов проверок в течение заданного промежутка времени, выдает человеку-оператору сигнал «Сменить начальный модуль».

Как видно из этого описания, ДМ реализует управление запусками и не затрагивает тех процедур, которые выполняются проверяющими модулями в процессе диагностирования компонентов ЦС со структурой симметричного двудольного графа. Поэтому приведенная в работе [2] организация диагностирования может быть использована для решения поставленной задачи.

3. ПРИМЕР

Особенности предлагаемого подхода к диагностированию возникшего отказа покажем на примере ЦС со структурой симметричного двудольного графа (система $Q_{7,3}$ на рисунке).

В цикле контроля, 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***.

В таблице T_S (табл.1) показана информация о составе проверяемых подсистем $U_{j,\,i,\,k}$ и номерах проверяющих их модулей m_j. В табл. 1 семь строк T_j, в каждой из которых для проверяющего модуля m_j, $j = 1, 2, ..., 7$, записаны номера компонентов, входящих в состав трех подсистем $U_{j,\,i1,\,k1}$, $U_{j,\,i2,\,k2}$, $U_{j,\,i3,\,k3}$, проверяемых модулем m_j, в ней также есть поля для записи оценок результатов проверки $r_{j,\,i1,\,k1}$, $r_{j,\,i2,\,k2}$, $r_{j,\,i3,\,k3}$ этих подсистем модулем m_j. (Отметим, что в клетках строк T_5, ..., T_3 табл. 1 показаны более компактные номера индексов проверяющего модуля и проверяемых подсистем.)

Действия исправного модуля m_j состоят в:

— исполнении функций проверяющего модуля в отношении связанных с ним проверяемых подсистем $U_{j,\,i1,\,k1}$, $U_{j,\,i2,\,k2}$, $U_{j,\,i3,\,k3}$,

— нахождении пути от одного исправного модуля к другому исправному для передачи ему функций проверяющего модуля.

Предположим, что из-за большого интервала между циклами контроля в рассматриваемой ЦС накопились отказы модуля 8 и двух линий связи (*4—13*), (*5—12*); кроме того, в процессе диагностирования *возник отказ модуля 6*.

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 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 передать результаты проверок в модуль *5*». Получив эту команду, модуль *7* (*4*, *6*) проводит самоконтроль подсистемы $U_{5,12,7}$ ($U_{5,9,4}$, $U_{5,14,6}$) и передает полученные результаты проверок модулю *5*.

Модуль *5* формирует оценки $r_{5,12,7} = 1$, $r_{5,9,4} = 0$, $r_{5,14,6} = 0$, записывает их в соответствующие клетки строки T_5 (табл. 2), анализирует полученные оценки: так как среди полученных оценок есть одна единичная и две нулевых, то передает модулю *4* подсистемы $U_{5,9,4}$ управление для продолжения процесса контроля. (Отметим, что в строке T_5 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 передать результаты проверок в модуль 4»;

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

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

— по нулевым оценкам в строке T_4 выделит индекс исправной подсистемы $U_{4,11,6}$, в котором найдет номер исправного модуля $\{q_4 = 6\}$, и модулю *6* передаст таблицу PT_4 и управление процессом диагностирования.

Результаты действий модуля *6* в качестве проверяющего модуля показаны в строке T_6 таблицы PT_4. Предположим, что после того, как модуль *6* проверил своих преемников и передал таблицу PT_4 и управление модулю *3*, *модуль 6 отказал*, что обнаружится при его проверке модулем *7* (строка T_7 в таблице PT_4).

Модуль *3* выполнит аналогичные действия, получит результаты проверок своих «преемников» — подсистем

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*	

$U_{3,14,5}$, $U_{3,13,4}$, $U_{3,8,2}$, выяснит, что из этих подсистем только одна исправна — $U_{3,14,5}$, и передаст модулю *5* функции проверяющего на следующем этапе.

Модуль *5* выполнит аналогичные действия, получит результаты проверок своих «преемников» — подсистем $U_{5,12,7}$, $U_{5,9,4}$, $U_{5,14,6}$, выяснит, что из этих подсистем только одна исправна — $U_{5,9,4}$ с модулем *4*. Но так как модуль *4* уже исполнял функции проверяющего модуля на этапе диагностирования, то модуль *5* должен искать другой исправный модуль для исполнения функций проверяющего среди ранее проверенных. В данном случае в строке T_6 есть нулевые оценки проверок модулей *2* и *1*, еще не исполнявших функции проверяющих. Таким образом, модуль *5* через модуль *12* должен передать модулю *2* таблицу PT_4 и управление процессом диагностирования.

Модуль *2* выполнит действия, аналогичные действиям модуля *5*, выяснит, что среди проверенных им подсистем есть только одна исправная — $U_{2,11,4}$, в которую входит модуль *4*, ранее исполнявший функции проверяющего. После этого модуль *2* найдет в строке T_6 исправный модуль *1*, ранее не использовавшийся, и передаст ему через модуль *8* таблицу PT_4 и управление дальнейшим процессом диагностирования.

Модуль *1*, выполнив аналогичные действия, найдет среди проверяемых им подсистем исправную подсистему $U_{1,10,7}$ и передаст модулю *7* таблицу PT_4 и управление процессом диагностирования.

Модуль *7* выполнит необходимые проверки подсистем $U_{7,10,6}$, $U_{7,12,2}$, $U_{7,13,3}$, определит, что среди его «преемников» есть два исправных модуля: *2* и *3*, которые ранее уже исполняли функции проверяющих. Поэтому, как и модули *5* и *2*, модуль *7* будет искать исправный модуль для исполнения функций проверяющего. Не обнаружив такого модуля в вышележащих строках таблицы PT_4, модуль *7* закончит процесс диагностирования и перешлет в ДМ таблицу PT_4 для обработки полученных результатов проверок.

Диагностический монитор, получив результаты первого запуска, начальным модулем второго запуска процесса диагностирования назначит модуль *7*, который был последним в цепочке проверяющих модулей первого запуска.

Таблица 2

Информация к примеру

Строка T_j	Проверяющий модуль	Проверяемые подсистемы $U_{j,i,k}$, оценки их проверок					
	m_j	$U_{j,i1,k1}$	$r_{j,i1,k1}$	$U_{j,i2,k2}$	$r_{j,i2,k2}$	$U_{j,i3,k3}$	$r_{j,i3,k3}$
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*	0
T_4	*4*	(*4-13*), *13*, (*13-3*), *3*	1	(*4-11*), *11*, (*11-6*), *6*	0	(*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*	**0**	(*4-9*), *9*, (*9-1*), *1*	0
T_6	*6*	(*6-14*), *14*, (*14-3*), *3*	**0**	(*6-11*), *11*, (*11-2*), *2*	0	(*6-10*), *10*, (*10-1*), *1*	0
T_3	*3*	(*3-14*), *14*, (*14-5*), *5*	**0**	(*3-13*), *13*, (*13-4*), *4*	1	(*3-8*), *8*, (*8-2*), *2*	1
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_2	*2*	(*2-11*), *11*, (*11-4*), *4*	0	(*2-8*), *8*, (*8-1*), *1*	1	(*2-12*), *12*, (*12-5*), *5*	1
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_7	*7*	(*7-10*), *10*, (*10-6*), *6*	1	(*7-12*), *12*, (*12-2*), *2*	0	(*7-13*), *13*, (*13-3*), *3*	0
Таблица PT_7							
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_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_3	*3*	(*3-14*), *14*, (*14-5*), *5*	**0**	(*3-13*), *13*, (*13-4*), *4*	1	(*3-8*), *8*, (*8-2*), *2*	1
Таблица PT_3							
T_3	*3*	(*3-14*), *14*, (*14-5*), *5*	**0**	(*3-13*), *13*, (*13-4*), *4*	1	(*3-8*), *8*, (*8-2*), *2*	1
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**
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_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

Модуль *7* выполнит действия, аналогичные действиям модуля 4 в начале первого запуска процесса диагностирования:

— образует промежуточную таблицу PT_7 и перепишет в нее из таблицы PT_4 строку T_7 с полученными оценками результатов проверок;

— по нулевым оценкам в строке T_7 выделит номера исправных преемников $\{q_7 = 2, 3\}$, и передаст одному из них (например, модулю *2*) таблицу PT_7 и управление процессом диагностирования.

Дальнейшие действия модуля *2* (*4*, *1*, *5*, *3*) в качестве проверяющего аналогичны его действиям в первом запуске. Результаты проверок, полученные модулем *2* (*4*, *1*, *5*, *3*), показаны в соответствующей строке таблицы PT_7. Модуль *3*, выполнив свои функции, выяснит, что все обнаруженные исправные модули уже исполняли функции проверяющих, перешлет в ДМ таблицу PT_7 для обработки полученных результатов проверок.

Сравнив информацию в таблицах PT_4 и PT_7, ДМ определит их различие по числу строк и запустит процесс диагностирования в третий раз, используя в качестве начального модуль *3* и в качестве исходной таблицу PT_7. Модуль *3* выполнит действия, аналогичные действиям модуля *7* в начале второго запуска процесса диагностирования, и передаст управление и таблицу PT_3 модулю *5*. Результаты проверок, полученные модулем *5* (*4*, *1*, *7*, *2*), показаны в соответствующей строке таблицы PT_3. Модуль *2*, выполнив свои функции, выяснит, что все обнаруженные исправные модули уже исполняли функции проверяющих, перешлет в ДМ таблицу PT_3 для обработки полученных результатов проверок.

После завершения третьего запуска ДМ, сравнив таблицы PT_7 и PT_3, обнаружит их совпадение как по номерам строк, так и по результатам проверок в этих строках и перейдет к обработке полученных результатов.

Процедура обработки полученных результатов проверок достаточно полно изложена в работе [2].

Для обработки и дешифрации результатов ДМ переписывает их из таблицы PT_3 в заранее заготовленную диагностическую таблицу T_D [2]. В клетках левого столбца таблицы T_D перечислены компоненты проверяемых подсистем $U_{j,\,i,\,k}$, взятые из соответствующих клеток таблицы состава T_S (см. табл. 1), а клетки следующего столбца предназначены для записи оценок $r_{j,\,i,\,k}$ проверок этих подсистем, полученных в процессе диагностирования. Последующие столбцы таблицы T_D обозначены символами одиночных компонентов системы $Q_{7,3}$: процессоров, блоков памяти, линий связи. В нижней части таблицы T_D находятся три строки, озаглавленные (в левом столбце) символами A_0, A_1, A_Z и предназначенные для записи результатов обработки полученной информации.

Пример в данном тексте подобран так, что после проявления отказа модуля 6 в системе создается отказовая ситуация, совпадающая с ситуацией, рассмотренной в Примере работы [2]. Это позволяет не приводить громоздкую таблицу T_D, подробности ее обработки и ограничиться изложением полученных результатов.

Расшифровка записей в строках A_0, A_1 и A_Z 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 подозреваются в отказе процессор *6*, 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_Z остались неопределенными состояния линий (*10—6*), (*1—8*), (*2—8*), (*6—11*), (*3—8*) и (*6—14*), так как их проявления «маскируются» неисправным состоянием одной из вершин (6 или 8), которые они соединяют.

Таким образом, по результатам диагностирования идентифицированы состояния 29 из 35 компонентов системы $Q_{7,3}$. Эти результаты ДМ индицирует человеку-оператору для проведения необходимых восстановительных процедур.

ЗАКЛЮЧЕНИЕ

Представленный подход к диагностированию возникшего отказа в цифровой системе со структурой симметричного двудольного графа основан на использовании особенностей метода, изложенного в работе [3]: для обнаружения таких отказов и выработки правильного диагноза выполняется дву- и более кратный запуск процесса диагностирования со сравнением полученных результатов проверок. Как и в работе [2], на время проведения диагностирования разнотипные устройства объединяются в проверяющие подсистемы одинакового состава. Предполагается, что в системе есть диагностический монитор, который инициирует процессы контроля и диагностирования в анализируемой цифровой системе, обрабатывает полученные результаты проверок, индицирует полученные результаты человеку-оператору.

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.* Организация диагностирования цифровых систем со структурой симметричного двудольного графа // Проблемы управления. — 2009. — № 6. — С. 59—67.
3. *Ведешенков В.А.* Самодиагностирование возникающих отказов и их устранение в цифровых системах с реконфигурацией // Проблемы управления. — 2004. — № 3. — С. 48—61.

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) 334-75-90, ✉ vva@ipu.ru.