

УДК 681.32:519.713

И.Ш. НЕВЛЮДОВ, Е.П. ВТОРОВ, Б.А. ШОСТАК

Харьковский национальный университет радиоэлектроники, Украина

ОПТИМИЗАЦИЯ ПРОВЕРКИ МИКРОПРОЦЕССОРНЫХ СИСТЕМ МЕТОДОМ МОДИФИЦИРОВАННЫХ РАСТУЩИХ СЕТЕЙ

Рассмотрен новый метод диагностики многопроцессорных систем повышенной сложности. Приведены основные положения и характеристики нового метода. Проанализирована возможность формализации метода в среде VB.NET.

диагностика микропроцессорных систем, ситуационное управление, технологическое оборудование

Введение

Диагностика микропроцессорных модулей технологических систем представляет собой значительную трудность как на этапе разработки, так и на этапе эксплуатации, причем, если на этапе разработки создание специализированных программных и аппаратных средств для тестирования создаваемой микропроцессорной системы экономически оправдано, то на этапе эксплуатации создание таких средств не представляется возможным ввиду их большой трудоемкости и как следствие, большой стоимости.

Кроме того, даже при необходимости тестирования небольшого количества различных типов микропроцессорных модулей создание для каждого типа модулей отдельных программных и аппаратных средств тестирования экономически невыгодно. При этом наличие существующих унифицированных средств тестирования (таких, как логические и сигнатурные анализаторы, внутрисхемные эмуляторы) улучшает ситуацию не значительно.

Следовательно, необходимо применять специальные эвристические методы поиска неисправностей [1], причем такой поиск значительно повышает производительность труда и, естественно, снижает требования, предъявляемые к уровню квалификации наладчиков.

Постановка проблемы

Анализ последних исследований показывает, что при диагностике микропроцессорных модулей минимизировать время диагностирования можно посредством применения экспертных оценок косвенных параметров сигналов в контрольных точках модуля. Контрольные точки микропроцессорных модулей систем ЧПУ технологического оборудования определяются, исходя из экспертных оценок приоритетности. При таком подходе возникает проблема создания оптимальной системы принятия решений (о неисправности того или иного дискретного элемента). Под такой системой понимается система, которая минимизирует заданный критерий качества для данного динамического процесса при заданных ограничениях. В случае описанного ниже метода проверки микропроцессорных систем в качестве критерия качества выступает риск принятия неверного решения при заданном входном пространстве состояний, а ограничения заключаются в фиксированной структуре базовой сети.

В теории оптимизации для целей синтеза оптимальной системы управления все рассматриваемые задачи делятся на два класса – задачи синтеза для малых областей пространства и задачи синтеза для больших областей пространства. Процедура синтеза для малых областей пространства состояний налага-

ет жесткие ограничения на решаемые задачи. В этом случае функция ошибки не является квадратичной, а уравнения динамического процесса нелинейны. В частности, рассматривается класс задач синтеза, определяемый критерием качества

$$q(t) = g[x(t), t], \quad (1)$$

уравнением выходного сигнала динамического процесса

$$q(t) = g[x(t), t], \quad (2)$$

уравнением состояния

$$x'(t) = f[x(t), m(t), t] \quad (3)$$

и ограничением управления

$$m(t) \in \mu(t). \quad (4)$$

Основные затруднения в задаче нелинейного оптимального управления встречаются при получении числового решения вариационной задачи. С одной стороны, ряд трудностей возникает при решении уравнений Понтрягина, а с другой стороны, трудности связаны с дискретной аппроксимацией получаемых уравнений динамического программирования [2].

Второй класс включает в себя задачи синтеза, в которых система работает в большой области пространства состояний. При этом границы области заданы и область не может быть представлена одной траекторией. Приложения, используемые для малых областей пространства состояний, не дают достаточной точности для этого класса задач. Поэтому в случае больших областей пространства состояний приходится прибегать к иным методам, один из которых описан ниже.

В дополнение к уже рассмотренным трудностям имеются еще два требования, которым должны удовлетворять методы аппроксимации. Во-первых, с увеличением сложности аппроксимации приближения должны равномерно сходиться к оптимальной системе управления. Во-вторых, когда аппроксимацию ограничивают некоторой степенью сложности, результирующая система должна быть устойчивой без нежелательных предельных циклов. Однако ни

ряды, ни какие-либо другие известные формы приближения не обладают этими свойствами.

Если рассматривать значения параметров сигналов микропроцессорных модулей в некотором пространстве дискретных состояний, то главная трудность синтеза системы управления процессом диагностики будет заключаться в несметном объеме данных, которые должны быть вычислены и преобразованы в соответствующее уравнения управления [3]. Необъятность этой задачи обусловлена наличием в пространстве состояний слишком большого количества точек, которые должны быть учтены в задаче. Однако в любой данный момент времени при диагностике микропроцессорных модулей представляет интерес только одна точка (в данном случае – один набор определенных параметров сигналов). Логический подход в таком случае состоит в создании системы управления, которая выполняет вычисления так, что работа системы в конечном итоге становится оптимальной при любом исходном состоянии системы. В качестве платы за такой подход приходится соглашаться с субоптимальными характеристиками системы в течение времени, необходимого для достижения оптимума [4, 5].

В основе предлагаемого метода лежит метод растущих пирамидальных сетей [6, 7]. Последовательная подача определенных входных наборов приводит к появлению рассогласования осциллограмм или сигнатур в определенных контрольных точках при различных входных наборах. Предположим, что есть определенный неисправный элемент в микропроцессорном модуле. Требуется на основе построения МРС для каждого электронного элемента выделить положительный контрольный элемент (если он активен, то данный элемент неисправен) и, соответственно, отрицательный контрольный элемент (если он активен, то данный элемент исправен).

Для этого необходимо определить понятия анализирующего элемента и регистратора. *Анализи-*

руючий елемент – гипотетический элемент, у которого нет реального физического аналога. Он представляет собой совокупность некоторых логических выводов (промежуточных или конечных в случае положительного или отрицательного контрольного элемента). *Регистратор* – реальная физическая цепь исследуемого модуля. Регистратор возбуждается, если на определенный цифровой вход подается определенный вид сигнала. Можно выделить три уровня (соответственно и три метода построения МРС):

- бинарные логические сигналы;
- значения сигналов, применяемые в шестизначном моделировании;
- определенная последовательность сигналов, распределенных во времени (или сигнатура).

Теперь необходимо сформулировать правила построения анализирующих элементов. Любой элемент МРС может быть в двух состояниях – возбужденном и невозбужденном. Набор возбужденных регистраторов в некоторый момент времени образуется в результате подачи на вход МРС в данный момент описания некоторого объекта через значения его признаков. Интервал между сменами описаний объектов, подаваемых на вход МРС, всегда превосходит интервал, в течение которого МРС возбужден. Это означает, что при подаче любого описания на регистраторы МРС находится в невозбужденном состоянии.

Если имеется некоторый анализирующий элемент, то *субмножество* этого элемента образуют все те элементы, от которых имеются пути, ведущие к входам данного элемента. Субмножество определяется вполне корректно, если учесть, что в МРС запрещены связи, которые могут привести к возникновению петель и циклов, иными словами, ни один анализирующий элемент не может быть соединен своим выходом со входом самого себя или входами каких-либо элементов, входящих в его субмножество. Нулевой слой субмножества некоторого эле-

мента образуют те элементы, выходы которых непосредственно поступают на входы данного элемента.

Аналогично можно определить и *супермножество* данного анализирующего элемента как множество тех элементов, к которым имеется путь, ведущий от выхода данного элемента. Те элементы, для которых этот выход непосредственно является входом, образуют нулевой слой супермножества.

Прежде всего необходимо выработать правила построения первоначальной (базовой) МРС для каждого ЛЭ. Формирование регистраторов происходит следующим образом. Для бинарной логики каждый вход цифровой системы имеет два регистратора: первый возбуждается, если на вход подается сигнал логического 0; второй – если подается сигнал логической 1. При этом соответственно, количество регистраторов определяется формулой

$$N = 2K + 2D, \quad (5)$$

где K – количество входов цифровой системы; D – количество контрольных точек.

При m -значном моделировании формула (5) будет выглядеть следующим образом:

$$N = M(K + D), \quad (6)$$

где M – значность логики.

Первоначально исследуем модель цифровой схемы, представленную на рис. 1, при бинарном коде активизации регистраторов.

Зная статические состояния регистраторов $R1 - R5$, контрольных точек $C4$ и $C5$ в определенный момент времени, необходимо построить МРС и выделить для каждого элемента $D1 - D5$ положительный контрольный (при отказе этого элемента) и отрицательный контрольный (при нормальном функционировании этого элемента) элементы, затем выполнить необходимый анализ возможности построения МРС и выделения отрицательных контрольных элементов понятия при всех исправных элементах $D1 - D5$. Кроме того, возможно построение МРС при различных видах отказов логических элементов. Соответственно, количество МРС в этих случаях прямо про-

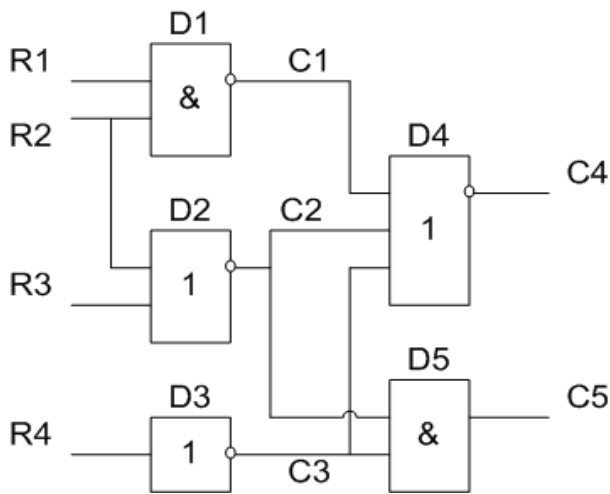


Рис. 1. Модель цифровой схемы

порционально количеству возможных неисправностей в каждом ЛЭ. Таким образом, возможно выделение положительных контрольных элементов не только по неисправным ЛЭ, но и по типам неисправностей.

Для определения неисправности элемента D4 формируется базовая сеть, представленная на рис. 2.

Процедура изменения MPC задается с помощью специальных правил:

1. Пусть при возбуждении некоторой комбинации регистраторов в существовавшей до этого MPC имелись анализирующие элементы, количество возбужденных элементов которых в нулевом слое субмножества удовлетворяет следующим условиям:

$$\begin{cases} V \geq 2; \\ N \neq K, \end{cases}$$

где N – количество возбужденных элементов субмножества; K – общее количество элементов субмножества.

Для всех таких элементов происходит следующее. Выходы всех возбужденных в нулевом слое субмножества элементов становятся входами нового анализирующего элемента, вошедшего в MPC, выход нового элемента становится входом исходного анализирующего элемента; все непосредственные связи, существовавшие у него до этого с возбужденными элементами из нулевого слоя субмножества,

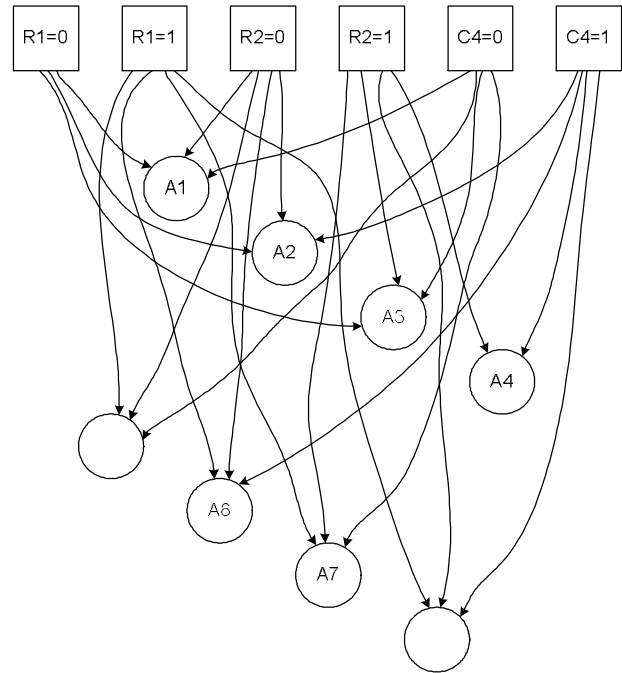


Рис. 2. Базовая MPC при неисправности элемента D4

ликвидируются. Выполнение этого правила иллюстрирует рис. 3. Те регистраторы, которые возбуждены, заштрихованы. В соответствии с правилом вводим новые элементы – A9, A10, A11.

После того, как в соответствии с правилом 1 введены все новые элементы, проверяются условия для применения правила 2, и если они удовлетворяются, то это правило реализуется.

2. Пусть при возбуждении некоторой комбинации регистраторов произошло возбуждение определенной части элементов в MPC. Если множество возбужденных элементов, в котором остались не возбужденными все элементы, входящие в нулевой слой их супермножества, состоит более чем из одного элемента, то вводится новый элемент, входами которого служат выходы всех упомянутых возбужденных элементов. Выполнение этого правила иллюстрируется ситуацией, изображенной на рис. 4. В соответствии с правилом вводим новый элемент – A12.

Перейдем теперь к процедуре определения принадлежности исследуемого элемента (цифровой микросхемы) к классу неисправных элементов с

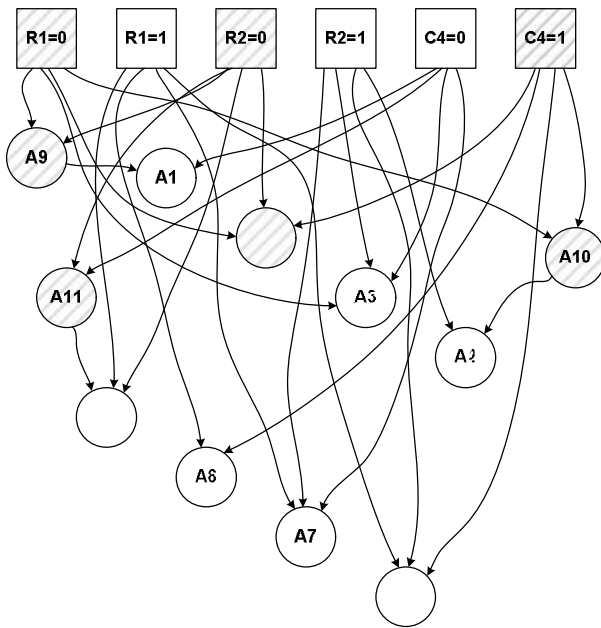


Рис. 3. MPC после применения правила 1

помощью MPC. В обучающей выборке с двоично-кодированными значениями признаков регистратор MPC, соответствующий данному значению некоторого признака, возбуждается, если это значение актуализируется, и не возбуждается в противном случае. На вход MPC последовательно подаются все значения признаков объектов из обучающей выборки как из группы положительных примеров, так и из группы отрицательных примеров. При подаче каждого значения происходит переформирование MPC в соответствии с описанными правилами.

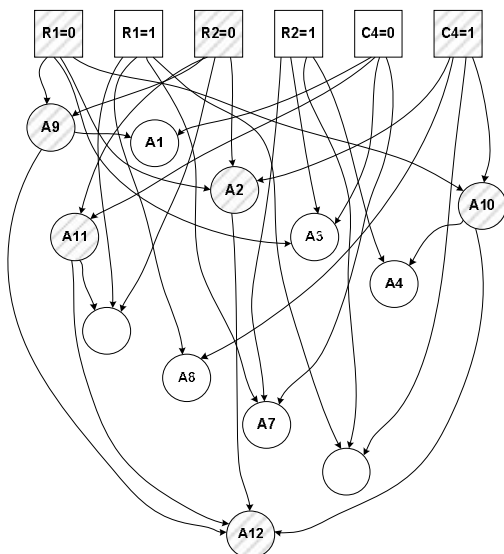


Рис. 4. MPC после применения правила 2

Каждому анализирующему элементу будем присывать два параметра m и n , значения которых могут меняться в процессе работы MPC. Параметр m – суммарное число возбуждений анализирующего элемента при подаче на регистраторы последовательности примеров. Параметр n характеризует число регистраторов, входящих в подмножество данного анализирующего элемента. В процессе преобразования MPC в множество ее анализирующих элементов выделяются специальные элементы, носящие название положительных и отрицательных контрольных элементов для исследуемой цифровой микросхемы.

Формирование понятий происходит за счет выполнения трех специальных процедур:

1. Если на регистраторы подается описание объекта обучающей выборки, который является положительным примером, и в MPC не выделен положительный контрольный элемент, то находится анализирующий элемент, у которого максимальны m и n . Такой элемент становится положительным контрольным элементом. (Активизация этого элемента свидетельствует о неисправности исследуемой микросхемы). Если в группе элементов с одинаковыми максимальными значениями m имеется несколько элементов с одинаковыми максимальными значениями n , то в качестве положительного контрольного элемента берется любой из них.

2. Если на регистраторы подается описание объекта обучающей выборки, который является отрицательным примером, и в MPC имеются положительные контрольные элементы, не содержащие в своих супермножествах других возбужденных контрольных элементов, то в каждом из этих супермножеств в качестве отрицательных контрольных элементов выбираются элементы, имеющие минимальное значение n , (активизация этого контрольного элемента свидетельствует об исправности исследуемой микросхемы). Если таких элементов несколько, то выбирается любой из них.

3. Если на регистраторы подается описание объекта обучающей выборки, который является положительным примером, и в MPC имеются отрицательные контрольные элементы, не содержащие в своих супермножествах других возбужденных контрольных элементов, то в каждом из этих супермножеств в качестве положительных контрольных элементов выбираются элементы, имеющие минимальное значение n . Если таких элементов несколько, то выбирается любой из них.

Контрольные элементы играют важную роль в процессе принятия решений о неисправности той или иной микросхемы. С помощью положительных контрольных элементов выделяются те сочетания регистраторов (значений признаков), которые с частотой m встречались в обучающей выборке на положительных примерах. С помощью отрицательных контрольных элементов выделяются сочетания значений признаков объектов, входящих в группу отрицательных примеров обучающей выборки. Но не для всех таких объектов, а только для тех, в которых сочетание значений признаков возбуждает положительные контрольные элементы, что соответствует наличию у отрицательного примера такого сочетания значений признаков, которое до этого было выделено в качестве важного (контролируемого) для формирования правила отнесения объектов к формируемому классу.

При формировании MPC обучающая выборка используется многократно. Как только в MPC в соответствии с одной из описанных выше процедур происходит появление нового контрольного элемента, начинается новый просмотр примеров из обучающей выборки, начиная с первого. Обучение считается завершенным, если описания всех объектов, входящих в обучающую выборку, при подаче их на регистраторы MPC не вызывают появления новых контрольных элементов.

В качестве обучающих выборок представленной выше модели было сформировано несколько шестизначных массивов значений регистраторов

(табл. 1). В результате роста MPC при подаче каждой обучающей выборки получим сеть, представленную на рис. 5 и в табл. 2. Заштрихованные ячейки соответствуют возбужденным элементам последнего набора обучающих выборок.

Таблица 1

Значения обучающих выборок

$R1(0)$	$R1(1)$	$R2(0)$	$R2(1)$	$C4(0)$	$C4(1)$	Примечание
1	0	1	0	1	0	отрицательная
1	0	1	0	0	1	положительная
1	0	0	1	0	1	положительная
0	1	1	0	1	0	отрицательная
0	1	1	0	0	1	положительная
0	1	0	1	0	1	положительная

После формирования MPC были определены положительный контрольный элемент $A04$ и отрицательный контрольный элемент $A12$. Последующие обучающие выборки не привели к появлению новых контрольных элементов. Когда понятие сформировано, MPC дает возможность пользоваться следующим правилом отнесения исследуемого объекта к классу неисправных элементов.

Объект a_i входит в множество неисправных элементов, если при введении состояний контрольных точек исследуемого модуля в регистраторы MPC в ней нет возбужденных отрицательных контрольных элементов, в супермножестве которых не имелся бы хотя бы один возбужденный положительный контрольный элемент. Объект a_i не входит в данный класс, если при введении его описания в регистраторы MPC в ней нет возбужденных положительных контрольных элементов, которые не содержали бы в своих супермножествах хотя бы одного возбужденного отрицательного контрольного элемента. Если ни одно из условий, сформулированных выше, не выполняется, то относительно объекта a_i не высказывается окончательных утверждений.

Можно показать, что три процедуры перестройки MPC, которые используются в процессе работы с

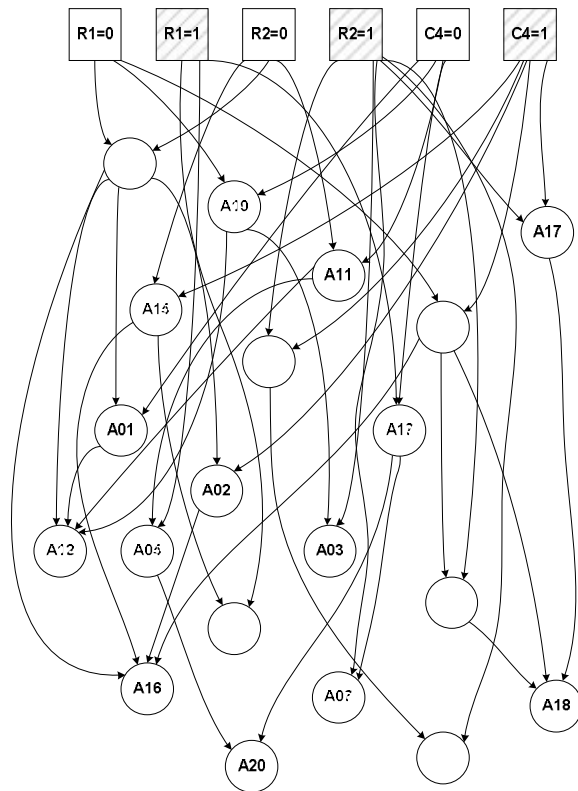


Рис. 5. MPC после подачи обучающих выборок

обучающей выборкой, всегда за конечное число шагов приводят к завершению формирования класса. При этом все положительные и отрицательные примеры из обучающей выборки распознаются MPC правильно.

При диагностике микропроцессорных систем весьма часто встречается ситуация, при которой из-за неполноты знаний об объекте и алгоритмах функционирования обучающая выборка может быть противоречивой. В этом случае одни и те же описания в ней могут входить и в группу положительных примеров, и в группу отрицательных примеров. Этот важный случай может быть реализован в методе MPC с помощью некоторой модификации процедуры ее построения.

В силу противоречивости обучающей выборки решение о включении или не включении объекта a_i в сформированный класс должно быть недетерминированным. Эта недетерминированность должна отражать те частоты, с которыми в обучающей выборке объект a_i относится к множествам положительных и отрицательных примеров.

Для формализации процесса построения MPC необходимо сформировать сначала массив базовой MPC. Базовая MPC формируется как текстовый файл и имеет структуру, представленную на рис. 6 [8],

Таблица 2

MPC после подачи обучающих выборок

Название вершины	L	M	K	Q	Массивы подмножеств вершин			
A0001/0	2	0	1	3	C0004_0	A0009/0		
A0002/0	2	1	1	3	C0004_1	A0009/0		
A0003/0	2	1	0	3	R0002_1	A0010/0		
A0004/0	2	1	2	3	R0002_1	A0014/0		
A0005/0	2	1	1	3	R0001_1	A0011/0		
A0006/0	2	1	0	3	R0001_1	A0015/0		
A0007/0	2	1	0	3	R0002_1	A0019/1		
A0008/1	2	2	0	3	R0001_1	A0017/1		
A0009/0	2	0	2	2	R0001_0	R0002_0		
A0010/0	2	0	1	2	R0001_0	C0004_0		
A0011/0	2	0	2	2	R0002_0	C0004_0		
A0012/0	4	0	1	3	A0001/0	A0009/0	A0010/0	A0011/1
A0014/0	2	1	2	2	R0001_0	C0004_1		
A0015/0	2	1	2	2	R0002_0	C0004_1		
A0016/0	4	0	1	4	A0002/0	A0009/0	A0014/0	A0015/0
A0017/1	2	2	1	2	R0001_1	C0004_1		
A0018/0	3	1	1	3	A0004/0	A0014/0	A0017/1	
A0019/1	2	1	1	2	R0001_1	C0004_0		
A0020/1	3	0	1	3	A0005/0	A0011/0	A0019/1	
A0021/0	2	0	1	3	A0006/0	A0015/0		

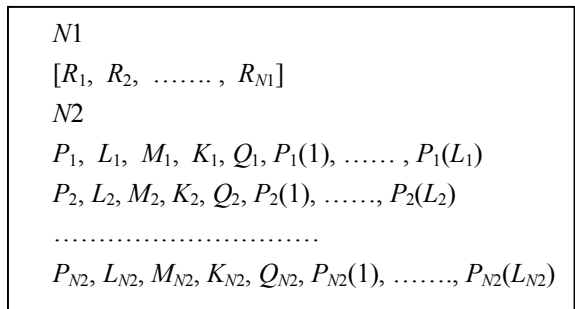


Рис. 6. Структура файла базовой MPC

где $N1$ – количество регистраторов; $[R_1, R_2, \dots, R_{N1}]$ – список названий регистраторов; $N2$ – количество анализирующих элементов MPC; P_{N2} – название анализирующего элемента; L_{N2} – количество элементов, входящих в нулевой слой подмножества данного анализирующего элемента; M_{N2} – количество воз-

бужденных элементов, входящих в нулевой слой субмножества данного анализирующего элемента; K_{N2} – количество возбуждений на предыдущих этапах; Q_{N2} – количество регистраторов, входящих в субмножество данного анализирующего элемента; $P_1(1) - P_{N2}(L_{N2})$ – названия вершин, входящих в субмножество данной вершины.

Обучающие выборки представляют собой одномерный вектор значений признаков. Значение 1 присваивается данному значению признака, если он возбужден, и 0 – в противном случае. Общая форма представления обучающих и экспертных выборок показана на рис. 7, где $V_1 - V_{M1}$ – текущие значения признаков; EXP – переменная, определяющая тип выборки (N – отрицательный пример, P – положительный пример, E – экспертная выборка).

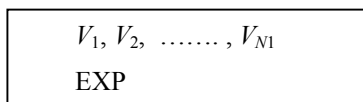


Рис. 7. Форма представления выборок

После завершения ввода положительных и отрицательных обучающих выборок формируется экспертная МРС. При подаче экспертной выборки на экспертную МРС происходит определение принадлежности неисправности к тому или иному классу отказов.

Заключение

Описанный выше метод диагностики многопроцессорных систем повышенной сложности обладает несколькими достоинствами. Во-первых, все выводы, формируемые с его помощью, допускают содержательную интерпретацию, так как задаются через логическую функцию, в которую в явном виде входят значения исходных признаков. Во-вторых, метод хорошо автоматизируется при работе на ЭВМ. В-третьих, после формирования окончательной МРС ту ее часть, которая не содержит контрольных элементов и не оказывает влияния на их возбуждение, можно удалить. При этом снижаются

требования к объему памяти ЭВМ (в данном программном обеспечении при объеме RAM 512 Мб максимальное количество вершин – 4000).

Литература

1. Малышенко Ю.В., Чипулис В.П., Шаршунов С.Г. Автоматизация диагностирования электронных устройств. – М.: Энергоатомиздат, 1986. – 216 с.
2. Мерриэм К.У. Теория оптимизации и расчет систем управления с обратной связью. – М.: Мир, 1967. – 549 с.
3. Джексон П. Введение в экспертные системы. – М.: Вильямс, 2001. – 622 с.
4. Горбунов В.Л., Панфилов Д.И., Преснухин Д.Л. Справочное пособие по микропроцессорам и микро-ЭВМ. – М.: Высш. шк., 1988. – 272 с.
5. Фергусон Дж., Макари Л., Уилльямз П. Обслуживание микропроцессорных систем. – М.: Мир, 1989. – 336 с.
6. Тейз А., Грибомон П., Луи Ж. Логический подход к искусственному интеллекту: от классической логики к логическому программированию: Пер. с франц. – М.: Мир, 1990. – 432 с.
7. Поспелов Д.А. Ситуационное управление: теория и практика. – М.: Наука, 1986. – 288 с.
8. Сайлер Б., Спотс Дж. Использование Visual Basic 6. – М.: Вильямс, 2000. – 830 с.
9. Корнелл Г., Моррисон Дж. Программирование на VB.NET. – С.-Пб.: Питер, 2002. – 394 с.
10. Стивен Р. Программирование в Win32 API на Visual Basic. – М.: ДМК, 2001. – 471 с.
11. Тоценко В.Г. Алгоритмы технического диагностирования дискретных устройств. – М.: Радио и связь, 1985. – 240 с.

Поступила в редакцию 21.04.2004

Рецензент: д-р техн. наук, зам. директора по науке А.Я. Мовшович, Научно-исследовательский институт технологии машиностроения, Харьков.