Организация группового контроля в дискретном устройстве

Г.П. Аксенова,
с.н.с, к.т.н., aksenova@ipu.ru
ИПУ РАН, Москва

Рассматривается множество контрольных точек в дискретном устройстве, в котором контроль организован по группам при помощи сигнатурного анализатора. Решается задача локализации неисправности любой кратности, находящейся в одной или в разных группах. Для этого дополнительно строятся различные разбиения множества контрольных точек, в которых организован такой же групповой контроль. Дан алгоритм построения дополнительных разбиений. Решается вопрос минимизации требуемой для этого контролирующей аппаратуры.

We consider a net of observation points in which a testing is organized by groups by means of fault signature analysis. We solve the problem of a localization of multiple faults when faults are in one or different groups. For this aim, we construct additional decompositions of a net, where a group testing is organized too. We propose the algorithm for constructing of additional decompositions. This method reduces the cost of testing hardware.

Введение

Проектирование отказоустойчивых систем опирается на широкое использование в них диагностирования с целью определения технического состояния объекта и указания места неисправных компонент. В режиме диагностирования исходный объект приводится к контролепригодному виду, т.е. в нем проводится декомпозиция, в результате чего он распадается на блоки (подсхемы), имеющие управляемые входы (на которые будет подаваться тестовая последовательность) и наблюдаемые выходы (контрольные точки), с которых снимается тестовая реакция. В качестве аппаратуры, анализирующей тестовую реакцию, обычно используют сигнатурные анализаторы (СА), одноканальные (одновходовые) и многоканальные (многовходовые) [1]. СА преобразует длинную выходную реакцию, получаемую с контрольных точек, в короткое ключевое слово – сигнатуру. Полученная сигнатура сравнивается с эталонной; на основании сравнения вырабатывается сигнал “0” (правильно) или “1” (неправильно).

Одноканальный СА анализирует реакцию, снимаемую только с одной контрольной точки тестируемого устройства. Так как контрольных точек обычно достаточно много, то выгоднее применять многоканальный СА, который, имея такой же объем аппаратуры, что и одноканальный СА, может анализировать реакцию сразу *n* контрольных точек, где *n* - число входов (разрядов) в СА. Для задачи проверки исправности объекта в целом использование многоканального СА имеет преимущество по аппаратурным затратам. Однако если надо указать, какая именно контрольная точка неисправна, то это невозможно сделать даже для одиночной неисправности. Отсюда возникает задача: как в случае использования многоканальных СА при неисправности любой кратности найти (локализовать) неисправную контрольную точку, причем с меньшими аппаратурными затратами, чем при использовании одноканальных СА.

Решению этой задачи посвящены работы автора [2, 3], где был предложен матричный метод локализации одиночной неисправности, а затем метод расширен на класс двух- и трехкратных неисправностей. Вопрос о локализации неисправностей большей кратности остался открытым, так как при этом геометрическое представление в матрице теряет наглядность, что затрудняет делать какие-нибудь выводы. В данном докладе этот вопрос рассматривается на языке теории множеств.

1. Постановка задачи

Дано: множество контрольных точек (к.т.), которое обозначим через *M*. Каждая к.т. подсоединяется к одному из входов какого-нибудь многоканального СА. Для всего множества потребуется $\{M\}/n$ СА, где $\{M\}$- мощность множества к.т., *n* - длина СА (число его входов), *n*≥16. Пока (для простоты) будем считать, что {*M*} кратно *n.* СА будем обозначать индексами из букв алфавита: $СА_A$, $СА_B$, $СА_C$, …

В результате такого подключения множество *M* разобьется на непересекающиеся подмножества, которые запишем в следующем виде:

$$\begin{gathered} A=\{a_1,a_2,\dots,a_n\}, \\ B=\{b_1,b_2,\dots,b_n\}, \\ C=\{c_1,c_2,\dots,c_n\}, \\ \dots\dots\dots\dots\dots\dots \end{gathered} \tag{1}$$

Здесь подмножества обозначены индексами подключенных к ним СА, а элементами этих подмножеств являются к.т. Контроль, организованный таким образом, будем называть групповым.

Объект исправен, если ни один СА не зафиксировал неисправность. Объект неисправен, если хотя бы один СА зафиксировал неисправность, однако при этом нельзя указать, какая именно к.т. неисправна, даже для одиночной неисправности. Групповой контроль определяет (локализует) неисправность только с точностью до подмножества.

Требуется: усовершенствовать групповой контроль так, чтобы с минимальными дополнительными аппаратурными затратами локализовать неисправные элементы во множестве *M*.

2. Локализация одиночных неисправностей

Сначала рассмотрим одиночные неисправности во множестве *M*, в котором число подмножеств равно числу элементов *n* в подмножестве. Разбиение ($n \times n$) будем называть квадратным, а множество обозначать как $M(n \times n)$.

Пусть неисправность зафиксировал один СА, например CA_B. Значит, неисправный элемент находится в подмножестве *B*. Следовательно, все элементы остальных подмножеств исправны. И если взять какой-нибудь элемент из подмножества *B*, например b_i, и проверить его в составе исправных элементов, взятых из других подмножеств, с помощью добавочного СА, то можно определить техническое состояние b_i, а именно: если СА не зафиксирует неисправность, значит, b_i исправен; в противном случае b_i неисправен. Обратим особое внимание, что при отборе элементов в дополнительное подмножество для проверки b_i воспрещается брать его соседей по подмножеству *B*, потому что неизвестно, какой сосед исправен, а какой нет.

Так как подмножество *B* содержит *n* элементов, то описанную процедуру надо провести для каждого элемента b_1, …, b_i, …, b_n. Будем брать в дополнительное подмножество для каждого элемента b_i по одному элементу из подмножеств *A, B, C,* … исходного разбиения. В результате получится *n* дополнительных подмножеств. Они образуют другое разбиение множества *M*, которое назовем первым дополнительным разбиением Δ1. Подмножества, входящие в Δ1, будем обозначать как $\Delta 1_1$, …, $\Delta 1_i$, …, $\Delta 1_n$.

П р и м е р 1. Если в подмножество $\Delta 1_i$, i=1, …, *n* набирать из подмножеств *A, B, C,*… элементы с одинаковыми индексами, то получим разбиение Δ1:

$$\Delta 1_1 = \{a_1, b_1, c_1, \dots\},\quad \Delta 1_2 = \{a_2, b_2, c_2, \dots\},\quad \Delta 1_3 = \{a_3, b_3, c_3, \dots\},\quad \dots\dots \tag{2}$$

Это разбиение в матричном представлении интерпретируется как столбцы (если исходное разбиение (1) интерпретировать как строки).

С учетом того, что одиночная неисправность может находиться в любом месте множества *M*, т.е. в любом подмножестве исходного разбиения (1), ту же процедуру следует провести для каждого элемента каждого подмножества *A, B, C,* …

Однако если при построении дополнительного разбиения Δ1 для одного исходного подмножества (выше строилось для подмножества *B*) следовать определенным принципам, то Δ1 будет служить дополнительным разбиением для любого элемента множества *M*.

Принципы отбора элементов в дополнительное подмножество:

№ 1. Из каждого исходного подмножества брать только один элемент. Другими словами, в подмножествах строящегося разбиения Δ1 не должны встречаться пары из одинаковых букв (a_i, a_j), (b_i, b_j), (c_i, c_j), … , i=1,…, *n*; j=1,…, *n*.

№ 2. Из каждого исходного подмножества брать один элемент только один раз (т.е. элемент, взятый в подмножество для элемента e_i, i=1,…, *n*, не должен выбираться для элемента e_j, j=1,…, *n*). Это означает, что а) разбиение Δ1 будет охватывать все элементы множества $M(n \times n)$ и б) подмножества в Δ1 не будут пересекаться между собой.

Отсюда следует

Т е о р е м а 1. Дополнительное разбиение множества M ($n \times n$)*, построенное по принципам № 1 и № 2 для элементов одного исходного подмножества, служит дополнительным разбиением для элементов всех остальных исходных подмножеств множества* $M(n \times n)$.

Д о к а з а т е л ь с т в о. Принципы отбора элементов в дополнительное подмножество обеспечивают, чтобы каждый элемент входил только в одно подмножество разбиения Δ1, точно так же как он входит только в одно исходное подмножество. Тем самым элемент является единственной точкой пересечения этих подмножеств. В составе этих двух подмножеств элемент и будет проверяться.

Сформулируем утверждение, доказательство которого вытекает из приведенных выше рассуждений.

У т в е р ж д е н и е 1. Групповой контроль, проведенный одновременно в двух разбиениях (исходном и дополнительном), локализует любую одиночную неисправность во множестве M. Если оба подмножества, куда входит элемент, зафиксированы неисправными, то элемент неисправен. Если хотя бы одно из этих подмножеств исправно, то элемент исправен (а второе подмножество неисправно как раз из-за одиночной неисправности, которая ищется).

При такой организации группового контроля аппаратурные затраты увеличиваются вдвое по сравнению с контролем по одному разбиению.

3. Локализация двукратных неисправностей

Теперь рассмотрим двукратные неисправности элементов из *M*. В этом случае при контроле исходного разбиения неисправность зафиксируют один или два СА (в зависимости от расположения неисправных элементов).

Если оба неисправных элемента находятся в одном неисправном исходном подмножестве, значит, все остальные подмножества исправны, их элементы можно использовать для построения дополнительных подмножеств так же, как и при одиночной неисправности, и неисправные элементы будут локализованы при одном дополнительном разбиении Δ1.

Если неисправные элементы находятся в разных подмножествах исходного разбиения (далее в статье везде будем подразумевать этот случай как более сложный), то при контроле будут зафиксированы неисправными два подмножества. Рассмотрим одно из них. Чтобы проверить, какие элементы в нем исправны, а какие нет, надо для каждого его элемента построить дополнительное подмножество из заведомо исправных элементов. Но этого сделать нельзя, потому что неизвестно, где находится вторая неисправность. Тогда построим для каждого элемента неисправного исходного подмножества не одно, а два дополнительных подмножества, не пересекающихся друг с другом (кроме исходного элемента). Если в первое дополнительное подмножество попал второй неисправный элемент, то во второе – уж точно нет. Это означает, что каждый элемент будет проверен хотя бы один раз в подмножестве с исправными соседями. Если оба дополнительных подмножества, куда входит элемент, будут зафиксированы неисправными, то, значит, виноват сам элемент, он неисправен. Если хотя бы одно из этих подмножеств исправно, то элемент исправен.

Как строить второе дополнительное подмножество $\Delta 2_e$ для произвольного элемента *e*? Естественно, следуя принципам отбора № 1 и № 2. Однако правила отбора теперь ужесточились: это подмножество не должно пересекаться с уже построенным для элемента *e* дополнительным подмножеством из разбиения Δ1 более чем в одном этом элементе *e*. Только тогда элемент *e* при проверке в составе подмножества $\Delta 2_e$ будет иметь других соседей, чем были раньше.

Например, если в качестве разбиения Δ1 взять разбиение (2), то в подмножествах строящегося разбиения Δ2 не должны встречаться пары с одинаковыми индексами (a_i, b_i), (b_j, c_j), (c_i, d_i), … , i=1,…, n; j=1,…, n. Это означает, что взято не более одного элемента из одного столбца.

Следовательно, к принципам отбора № 1 и № 2 добавится еще один.

Принцип отбора № 3 (для разбиения Δ2). При построении для элемента *e* дополнительного подмножества $\Delta 2_e$ запрещается брать элементы, взятые в уже построенное подмножество $\Delta 1_e$.

П р и м е р 2. Продолжим пример 1 и построим для него разбиение Δ2. Разбиение будем строить для исходного подмножества *A*. Для элемента a_1 подмножество $\Delta 2_1$ будет выглядеть так: a_1 ,b_2 ,c_3 ,d_4 ,e_5,… Элемент b_2 взят как следующий за b_1, потому что b_1 брать нельзя: он входит в подмножество $\Delta 1_1$. Элемент c_3 взят потому, что c_1 и c_2 брать нельзя, иначе будут два пересечения с уже построенными подмножествами $\Delta 1_1$ и $\Delta 1_2$, т.е. будут пары с одинаковыми индексами (a_1, c_1) и (b_2, c_2). И так далее. Аналогично можно построить подмножества $\Delta 2_2$, $\Delta 2_3$, …, беря элементы из исходных подмножеств справа от только что взятых. Получим разбиение Δ2:

(3)
$$\Delta 2_1 = \{a_1, b_2, c_3, d_4, e_5, \dots\},$$
$$\Delta 2_2 = \{a_2, b_3, c_4, d_5, e_6, \dots\},$$
$$\Delta 2_3 = \{a_3, b_4, c_5, d_6, e_7, \dots\},$$
$$\dots\dots\dots\dots\dots\dots\dots\dots,$$

которое в матричном представлении интерпретируется как диагонали.

Из примера видно, что если взять любой элемент, скажем c_3 из подмножества *C*, то он входит только в одно подмножество $\Delta 1_3$ и только в одно подмножество $\Delta 2_1$. Кроме того, видно, что подмножества в Δ2 по индексам параллельно сдвинуты относительно друг друга, поэтому и не пересекаются.

4. Локализация неисправностей кратности p≥2

Из предыдущего раздела становится ясно, как локализовать неисправность произвольной кратности p≥2. Для этого надо для каждого элемента построить *p* непересекающихся дополнительных подмножеств. Тогда каждый элемент будет проверяться в составе *p*+1 подмножествах (исходном и дополнительных). Если элемент неисправен, то все *p*+1 подмножеств будут зафиксированы неисправными (из-за этого элемента). Если элемент исправен, то при *p*-кратной неисправности могут быть зафиксированы неисправными *p* подмножеств, а одно уж точно будет исправно.

При построении дополнительных подмножеств для локализации *p*-кратной неисправности принцип отбора № 3, изложенный для разбиения Δ2, будет звучать уже в общем виде.

Принцип отбора № 3. *При построении для какого-либо элемента очередного дополнительного подмножества запрещается брать элементы, взятые в ранее построенные дополнительные подмножества для этого элемента.*

Другими словами, во всех построенных для элемента подмножествах не должны встречаться одинаковые пары остальных элементов.

На основе вышеизложенного можно сформулировать необходимые и достаточные условия локализации неисправности любой кратности при помощи группового контроля.

Т е о р е м а 2. Необходимым условием локализации неисправности любой кратности является построение хотя бы одного такого разбиения, в котором каждый элемент проверялся бы в составе с заведомо исправными элементами.

Т е о р е м а 3. Достаточным условием локализации p-кратной (1≤ p ≤ n) неисправности является построение для исходного разбиения (1) p дополнительных разбиений, удовлетворяющих принципам отбора № 1 – № 3.

Доказательство теорем вытекает из приведенных выше рассуждений.

5. Алгоритм построения дополнительных разбиений

Дополнительные разбиения можно строить методом случайного выбора элементов. Однако, как показали экспериментальные построения, на каком-то шаге может возникнуть ситуация, что нельзя найти следующий элемент в подмножество, удовлетворяющий принципам отбора, а вот если бы на каком-то предыдущем шаге был выбран другой элемент, то этой проблемы не возникло бы. Т.е. метод случайного выбора в общем случае требует полного перебора. Поэтому хорошо бы использовать какой-нибудь несложный алгоритм, даже если он имеет какие-то ограничения.

В этой связи предлагается алгоритм сдвига индексов (АСИ), заключающийся в том, что в следующем разбиении подбираются элементы, имеющие индекс больший, чем были в предыдущем разбиении, на некоторую величину. Выведем для этой величины формулу. Припишем исходным подмножествам *A, B, C*,… номера *j*, равные 0, 1, 2, … (номера показаны в скобках):

(4)
$$A = a_1a_2a_3a_4a_5a_6a_7 \qquad (0)$$
$$B = b_1b_2b_3b_4b_5b_6b_7 \qquad (1)$$
$$C = c_1c_2c_3c_4c_5c_6c_7 \qquad (2)$$
$$D = d_1d_2d_3d_4d_5d_6d_7 \qquad (3)$$
$$E = e_1e_2e_3e_4e_5e_6e_7 \qquad (4)$$
$$F = f_1f_2f_3f_4f_5f_6f_7 \qquad (5)$$

Подмножеству *A* в (4) дан номер *j*=0; это означает, что разбиения строятся для элементов именно этого подмножества.

Индексы элементов в подмножестве обозначим через *i*= 1, 2, …, *n*. Индекс элемента, который по алгоритму берется в соответствующее дополнительное подмножество, обозначим через I_j, где *j* указывает номер подмножества, из которого берется элемент,

$$I_j = (i + kj)(mod\ n). \qquad (5)$$

Здесь *k* – коэффициент сдвига, значение которого связано с номером строящегося разбиения *P* соотношением

$$k = P - 1. \qquad (6)$$

Например, какие элементы будут взяты в дополнительное подмножество $\Delta 5_3$? Номер подмножества указывает, что *P*=5; индекс при номере – что *i* =3, т.е. подмножество строится от элемента a_3. Вычисляем *k* по формуле (6): *k*=5–1=4. Значит, $I_j = i + 4j$ по формуле (5). Тогда из подмножества *j* =0 (т.е. из *A*) надо взять элемент с номером (3+4·0)=3 (т.е. a_3); из подмножества *j* =1 (т.е. из *B*) надо взять (3+4·1)-й элемент (т.е. b_7); из подмножества *j* =2 (т.е. из *C*) надо взять элемент с номером (3+4·2)=11 (т.е. c_{11}) и т.д.

Заметим, что в алгоритме сдвиг индексов производится по кольцу (циклически, т.е. за индексом *n* вновь идет 1); для этого вычисленное по формуле (5) значение индекса берется по *mod n*.

Первое разбиение Δ1, которое построит алгоритм АСИ, – это столбцы. А именно, номер этого разбиения *P* = 1, значит, *k* = 0 (т.е. нулевой сдвиг). Из (5) получим, что $I_j = i$, т.е. индексы I_j у элементов, которые берутся из всех исходных подмножеств *A, B, C, …* (их номера 0,1,2, … соответственно) равны *i* - номеру элемента a_i, из которого строится подмножество. Это означает, что в одном подмножестве разбиения Δ1 индексы у элементов одинаковы: в $\Delta 1_1$ – это «1», в $\Delta 1_2$ – это «2», в $\Delta 1_3$ – это «3» и т.д.

Второе разбиение Δ2, которое построит АСИ, – это диагонали, т.е. сдвиг на единицу (см. (3)). Последующие разбиения – это в матричном представлении «более наклонные» диагонали. Всего можно построить *n* разбиений (по числу элементов в исходном подмножестве А). Заметим, что при сдвиге на *n* +1 вновь получается разбиение «столбцы».

П р и м е р 3. Построим по алгоритму АСИ все разбиения для множества *M*(7×7):

	Δ1	Δ2	Δ3	Δ4
	$a_1b_1c_1d_1e_1f_1g_1$	$a_1b_2c_3d_4e_5f_6g_7$	$a_1b_3c_5d_7e_2f_4g_6$	$a_1b_4c_7d_3e_6f_2g_5$
	$a_2b_2c_2d_2e_2f_2g_2$	$a_2b_3c_4d_5e_6f_7g_1$	$a_2b_4c_6d_1e_3f_5g_7$	$a_2b_5c_1d_4e_7f_3g_6$
	$a_3b_3c_3d_3e_3f_3g_3$	$a_3b_4c_5d_6e_7f_1g_2$	$a_3b_5c_7d_2e_4f_6g_1$	$a_3b_6c_2d_5e_1f_4g_7$
	$a_4b_4c_4d_4e_4f_4g_4$	$a_4b_5c_6d_7e_1f_2g_3$	$a_4b_6c_1d_3e_5f_7g_2$	$a_4b_7c_3d_6e_2f_5g_1$
	$a_5b_5c_5d_5e_5f_5g_5$	$a_5b_6c_7d_1e_2f_3g_4$	$a_5b_7c_2d_4e_6f_1g_3$	$a_5b_1c_4d_7e_3f_6g_2$
(7)	$a_6b_6c_6d_6e_6f_6g_6$	$a_6b_7c_1d_2e_3f_4g_5$	$a_6b_1c_3d_5e_7f_2g_4$	$a_6b_2c_5d_1e_4f_7g_3$
	$a_7b_7c_7d_7e_7f_7g_7$	$a_7b_1c_2d_3e_4f_5g_6$	$a_7b_2c_4d_6e_1f_3g_5$	$a_7b_3c_6d_2e_5f_1g_4$

Δ5	Δ6	Δ7
$a_1b_5c_2d_6e_3f_7g_4$	$a_1b_6c_4d_2e_7f_5g_3$	$a_1b_7c_6d_5e_4f_3g_2$
$a_2b_6c_3d_7e_4f_1g_5$	$a_2b_7c_5d_3e_1f_6g_4$	$a_2b_1c_7d_6e_5f_4g_3$
$a_3b_7c_4d_1e_5f_2g_6$	$a_3b_1c_6d_4e_4f_7g_5$	$a_3b_2c_1d_7e_6f_5g_4$
$a_4b_1c_5d_2e_6f_3g_7$	$a_4b_2c_7d_5e_3f_1g_6$	$a_4b_3c_2d_1e_7f_6g_5$
$a_5b_2c_6d_3e_7f_4g_1$	$a_5b_3c_1d_6e_4f_2g_7$	$a_5b_4c_3d_2e_1f_7g_6$
$a_6b_3c_7d_4e_1f_5g_2$	$a_6b_4c_2d_7e_5f_3g_1$	$a_6b_5c_4d_3e_2f_1g_7$
$a_7b_4c_1d_5e_2f_6g_3$	$a_7b_5c_3d_1e_6f_4g_2$	$a_7b_6c_5d_4e_3f_2g_1$

Из (7) хорошо видно, как внутри любого разбиения алгоритм АСИ устраивает параллельный сдвиг индексов: параллельные столбцы, параллельные диагонали; этим обеспечивается непересекаемость подмножеств внутри разбиения.

Из (7) также видно, как легко можно строить эти разбиения. Построить надо только первое подмножество в каждом разбиении. Во втором подмножестве разбиения индексы у элементов надо увеличить на 1 по сравнению с первым; в третьем подмножестве – увеличить на 1 по сравнению со вторым и так до конца, увеличивая индексы на 1 по сравнению с предыдущим подмножеством. Получится, что в каждом разбиении идет последовательный перебор индексов у каждого элемента из разного начального состояния.

Алгоритм АСИ имеет ограничение, заключающееся в том, что если коэффициент сдвига индекса *k* имеет общий сомножитель с *n* - числом элементов, для которых строятся подмножества в разбиении, то построенное разбиение не удовлетворяет принципам отбора. Покажем это на примере с множеством *M*(6×6).

$$A = a_1a_2a_3a_4a_5a_6 \qquad (0)$$
$$B = b_1b_2b_3b_4b_5b_6 \qquad (1)$$
$$C = c_1c_2c_3c_4c_5c_6 \qquad (2)$$
$$D = d_1d_2d_3d_4d_5d_6 \qquad (3)$$
$$E = e_1e_2e_3e_4e_5e_6 \qquad (4)$$
$$F = f_1f_2f_3f_4f_5f_6 \qquad (5).$$

Построим для него дополнительные разбиения Δ1, …, Δ6:

	Δ1	Δ2	Δ3	Δ4	Δ5	Δ6
	$a_1b_1c_1d_1e_1f_1$	$a_1b_2c_3d_4e_5f_6$	$a_1b_3c_5d_1e_3f_5$	$a_1b_4c_1d_4e_1f_4$	$a_1b_5c_3d_1e_5f_3$	$a_1b_6c_5d_4e_3f_2$
	$a_2b_2c_2d_2e_2f_2$	$a_2b_3c_4d_5e_6f_1$	$a_2b_4c_6d_2e_4f_6$	$a_2b_5c_2d_5e_2f_5$	$a_2b_6c_4d_2e_6f_4$	$a_2b_1c_6d_5e_4f_3$
(8)	$a_3b_3c_3d_3e_3f_3$	$a_3b_4c_5d_6e_1f_2$	$a_3b_5c_1d_3e_5f_1$	$a_3b_6c_3d_6e_3f_6$	$a_3b_1c_5d_3e_1f_5$	$\square_3\square_2\square_1\square_6\square_5\square_4$
	$\square_4\square_4\square_4\square_4\square_4\square_4$	$\square_4\square_5\square_6\square_1\square_2\square_3$	$\square_4\square_6\square_2\square_4\square_6\square_2$	$\square_4\square_1\square_4\square_1\square_4\square_1$	$\square_4\square_2\square_6\square_4\square_2\square_6$	$\square_4\square_3\square_2\square_1\square_6\square_5$
	$\square_5\square_5\square_5\square_5\square_5\square_5$	$\square_5\square_6\square_1\square_2\square_3\square_4$	$\square_5\square_1\square_3\square_5\square_1\square_3$	$\square_5\square_2\square_5\square_2\square_5\square_2$	$\square_5\square_3\square_1\square_5\square_3\square_1$	$\square_5\square_4c_3d_2e_1f_6$
	$a_6b_6c_6d_6e_6f_6$	$a_6b_1c_2d_3e_4f_5$	$a_6b_2c_4d_6e_2f_4$	$a_6b_3c_6d_3e_6f_3$	$a_6b_4c_2d_6e_4f_2$	$a_6b_5c_4d_3e_2f_1$

Из (8) видно, что в разбиениях Δ3, Δ4, Δ5 (со сдвигами соответственно 2, 3, 4, т.е. числами, имеющими общий сомножитель с числом 6) выбор элемента опять попадает в исходный столбец, с которого начали, и далее индексы начинают повторяться - налицо пересечения подмножеств. Количество разбиений, удовлетворяющих принципам отбора, всего три (Δ1, Δ2, Δ6 со сдвигами соответственно 0, 1, 5), а с тремя дополнительными разбиениями можно локализовать неисправности максимум трехкратные. Повышать кратность в таких случаях предлагается способом, вытекающим из следующей теоремы.

Т е о р е м а 4. Для любого множества M' *всегда можно получить разбиение, удовлетворяющее принципам отбора, из разбиения, построенного по этим принципам для множества большей размерности* $M(n \times n)$*, где* $(n \times n) > \{M'\}$*, путем вычеркивания ненужных элементов.*

Доказательство и пример опускаются из-за недостатка места.

Все изложенное в предыдущих разделах доказывает

У т в е р ж д е н и е 2. Групповой контроль, проведенный одновременно в исходном и р дополнительных разбиениях, локализует во множестве М любую неисправность кратности не больше р. Если все подмножества, куда входит элемент, зафиксированы неисправными, то элемент неисправен. Если хотя бы одно из этих подмножеств исправно, то элемент исправен.

Следует отметить, что данная работа предоставляет возможность разработчику самому выбирать приемлемый вариант локализации неисправности: кратность, аппаратурные затраты, таблицы дополнительных разбиений, даже исходное разбиение (множество контрольных точек можно покрыть не обязательно одним исходным разбиением, а несколькими, удовлетворяющими разработчика по размерности, по длине подмножеств).

6. Аппаратурные затраты на групповой контроль

По алгоритму АСИ можно построить только *n* разбиений. А больше и не следует строить, потому что не получится выигрыш в контролирующей аппаратуре (в аппаратуре СА). Напомним, что *n* - число разрядов (триггеров) в СА. При контроле элементов в составе исходного разбиения на 1 элемент тратится 1 триггер СА и неисправность локализуется с точностью до группы; при контроле элементов в составе исходного и *р* дополнительных разбиений на 1 элемент тратится (*р*+1) триггеров из разных СА и локализуется *р*-кратная неисправность. А вот при контроле в составе *n* разбиений на 1 элемент тратится *n* триггеров из разных СА, т.е. как и в случае индивидуального контроля, когда в каждой контрольной точке используется свой одноканальный СА. Таким образом, при групповом контроле получается выигрыш в аппаратуре по сравнению с индивидуальным контролем тогда, когда для элемента строится не более чем (*n*-1) дополнительных разбиений. Если же требуется локализовать во множестве *М* неисправность кратности, большей *n*, то тогда следует увеличить саму длину СА до нужного значения.

При индивидуальном контроле для локализации неисправности любой кратности в множестве $M(n \times n)$ общие затраты на аппаратуру при *n*-разрядном СА составляют $(n \times n \times n) = n^3$ триггеров. Для локализации одиночной неисправности групповым контролем на СА потребуется $2(n \times n) = 2n^2$ триггеров. Следовательно, если *n* = 20, то затраты на аппаратуру при групповом контроле на порядок будут ниже ($n^3/_{2n^2} = n/_2 = 10$), чем при индивидуальном. По отношению к кратным неисправностям затраты на СА зависят от кратности *p* и составляют $(p + 1)n^2$ триггеров; по сравнению с индивидуальным контролем это в $n/_{(p+1)}$ раз меньше.

Заключение

В работе решена задача локализации неисправностей любой кратности в дискретном устройстве, представленном множеством своих контрольных точек. Это сделано за счет повышения «размерности» проверки элементов множества, т.е. увеличения числа подмножеств, в составе которых каждый элемент одновременно проверяется. Такая организация проверки элементов названа групповым контролем. Найдены необходимые и достаточные условия локализации неисправности любой кратности при групповом контроле. Дан алгоритм построения дополнительных подмножеств для проверки элементов, так чтобы их число было минимально и чтобы в них проверялся каждый элемент из множества контрольных точек.

Предложенный групповой контроль обеспечивает сокращение аппаратурных затрат на локализацию неисправностей по сравнению с индивидуальным контролем: для одиночных неисправностей на порядок или больше (в зависимости от выбранной длины сигнатурного анализатора), для кратных неисправностей - в зависимости от кратности. И лишь при кратности, равной длине сигнатурного анализатора, объемы затрат совпадают. Поставленная в статье задача решена.

Литература

1. Ярмолик В.Н. Контроль и диагностика цифровых узлов ЭВМ. Минск: Наука и техника, 1988.
2. Аксенова Г.П. Матричный способ локализации неисправностей в ПЛИС // АиТ. 2013. № 9. С. 119 - 124.
   Aksenova G.P. A Matrix Method for PLD Failure Localization // Autom. Remote Control. 2013. V. 74. No. 9. P. 1525 – 1529.
3. Аксенова Г.П. Повышение разрешающей способности матричного метода локализации неисправностей // АиТ. 2016. № 8. С. 159 - 166.
   Aksenova G.P. Increasing Resolvability for the Matrix Fault Localization Method // Autom. Remote Control. 2016. V. 77. No. 8. P. 1447 – 1452.