УДК 651.326

Л.В. ДЕРБУНОВИЧ, д-р техн. наук, проф. НТУ «ХПИ»,
М.А. БЕРЕЖНАЯ, канд. техн. наук, доц. ХНУРЭ,
Я.Ю. КОРОЛЕВА, асс. ХНУРЭ, ***М.Г. РЫЖИКОВА,*** асс. ХНУРЭ

ТЕСТОВОЕ ДИАГНОСТИРОВАНИЕ ОДНОМЕРНЫХ ОДНОРОДНЫХ СТРУКТУР

Пропонується процедура синтезу перевіряючих тестів для виявлення класу функціональних несправностей в одномірних однорідних мережах (ОМ) по автоматним моделям осередка ОМ і знаходженню в них фундаментальних циклів. Також отримана оцінка трудомісткості процедури синтезу.

The test method of one-dimensional iterative logic arrays (ILAs), composed of identical cells, are considered. The fault model assumed is that faults in single cell can change a cell behavior in any arbitrary way. The method is based on finding fundamental circles in automaton model of ILA cell. The complexity of the test procedure is derived.

Введение. Широкое распространение СБИС, программируемых логических интегральных схем (ПЛИС) типа *FPGA* и *CPLD*, обладающих регулярностью структурной организации, определяет интерес исследователей к проблеме реализации схем дискретных устройств (ДУ) и систем в виде однородной системы или сети (ОС) из многофункциональных элементов или ячеек, каждая из которых повторяется вместе со своими связями или в виде структур клеточных автоматов [1, 2, 3]. Большое число работ в этой области связано с разработкой методов оптимального размещения автоматной модели ДУ в ОС. В качестве критерия оптимальности в большинстве случаев используется минимальность площади кристалла [4].

Другим критерием реализации ДУ в виде ОС является простота организации процедуры тестового диагностирования ОС. Решению проблемы тестового диагностирования ОС посвящен ряд работ отечественных и зарубежных авторов [5, 6, 7]. Определим терминологию, используемую в дальнейшем изложении, на примере одномерной ОС, представленной на рис. 1.

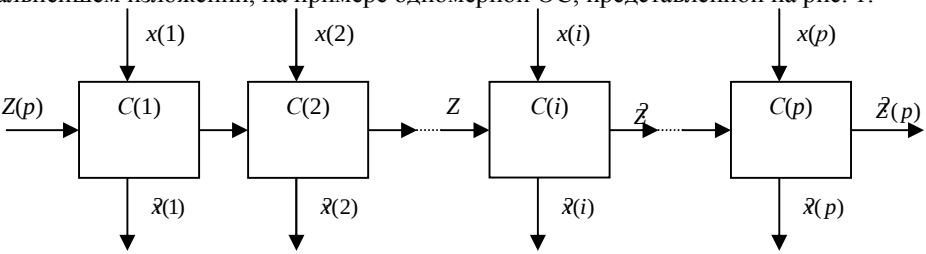

Рис. 1. Одномерная ОС с наблюдаемыми выходами $\hat{x}$

В такой сети левый вход z и все входы x являются управляемыми. Выходы $\hat{x}$ и выход $\hat{z}$ крайней правой ячейки являются наблюдаемыми. Предполагается, что переменная, запитывающая вход x_i ячейки $C(i)$ не зависит от всех других переменных, подаваемых на входы $x_j \neq x_i$. Поведение ячейки можно описать таблицей переходов-выходов (ТПВ) автоматной модели $C(i)$, в которой каждая строка кодируется переменными z (переменные состояния), а столбцы - переменными x. В клетках таблицы переходов записываются пары $(\hat{z}, \hat{x})$ для каждой комбинации входных переменных (z, x) ячейки сети. В общем случае, в зависимости от функции ячейки ОС, пары (z, x) и $(\hat{z}, \hat{x})$ могут представлять собой пары двоичных векторов различной размерности. Для обнаружения неисправности необходимо создать условия ее проявления и транспортировки на наблюдаемые выходы $\hat{z}$ и $\hat{x}$. Если для заданной сети эти условия определены значениями переменных x, подаваемых на верхние входы сети, и переменных z, подаваемых на крайний левый вход сети, то говорят, что сеть тестируема относительно установленного класса неисправностей.

Различают два типа одномерных однородных сетей: с наблюдаемыми выходами $\hat{x}$ и без них (рис. 1 и рис. 2).

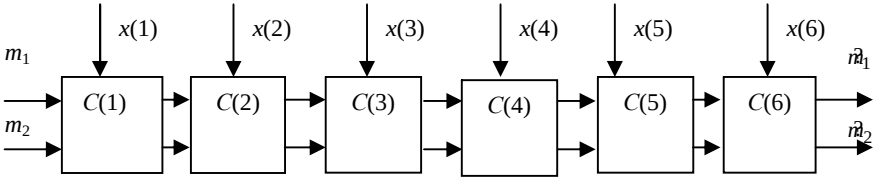

Рис. 2. Одномерная ОС без $\hat{x}$ выходов

В общем случае отсутствие наблюдаемых выходов $\hat{x}$ в сети усложняет процедуру проверки ее исправности ввиду необходимости транспортировки множества неисправностей каждой ячейки сети на крайний правый выход сети.

Принципы построения проверяющих тестов на функциональном уровне исследованы в работах [3, 4], в которых ячейка ОС рассматривается как совокупность четырех взаимодействующих каналов. Задача проверки исправности решается приложением совокупности совместимых, сопряженных и самосопряженных тестов-наборов в некотором *i*-ом направлении распространения сигналов, где «*i*» изменяется по всем выходам ячейки сети. Получение требуемых тестовых наборов на входах проверяемой ячейки осуществляется путем настройки соответствующих каналов ОС. К

сожалению, в этих работах не освещаются вопросы полноты проверяющего эксперимента, класс обнаруживаемых неисправностей и влияние структуры ячейки сети или ее функциональных характеристик на сложность построения и реализации диагностического эксперимента.

На уровне сети рассматриваются две модели неисправностей: 1) модель одиночной неисправности сети (допускается неисправной одна ячейка сети); 2) модель кратной неисправности сети (допускается неисправным произвольное множество ячеек сети). Первая модель представляет класс неисправностей F_1, которые изменяют (искажают) таблицу переходов-выходов автоматной модели ячейки сети при ограничении: неисправность не изменяет числа состояний ячейки, является устойчивой на время прохождения проверяющего теста и допускается неисправной в момент проверки лишь одна произвольная ячейка сети. Класс неисправностей F_1 включает полное множество константных неисправностей ячейки, подкласс перемычек и коротких замыканий, перепутываний и инверсий, не увеличивающих числа состояний ячейки.

Вторая модель кратной неисправности ячеек сети представляет класс неисправностей F_k, когда при тех же ограничениях на изменения автоматной диаграммы ячейки сети, которые определены для класса F_1, допускается неисправным произвольное множество ячеек сети.

В зависимости от свойств ОС различают сети, у которых длина проверяющих тестов постоянна и не зависит от числа ячеек сети. Такие ОС называют С - тестируемыми сетями или С-ОС. Если длина проверяемого теста линейно зависит от числа ячеек сети, то последние называют L - тестируемыми сетями или L -ОС.

Проблема тестового диагностирования одномерных ОС была исследована в работе [7], в которой были предложены методы тестирования неисправностей класса F_1 в L-ОС. В [8] определены необходимые и достаточные условия L-тестируемости одномерной однородной сети без наблюдаемых выходов $\hat{x}$ относительно класса одиночных неисправностей ячеек сети. Показано, что сеть является тестируемой относительно класса неисправностей F_1, если в таблице переходов ячейки множество последующих состояний содержит все состояния и автоматная модель ячейки является минимальным автоматом Мура, то есть в таблице переходов нет двух одинаковых строк. Для нахождения проверяющих тестов сети в [8] был предложен подход, основанный на построении тестового графа и выделении множества фундаментальных циклов в тестовом графе сети. Однако предложенный подход не доведен до уровня алгоритмической завершенности и отсутствует оценка трудоемкости процедуры синтеза проверяющих тестов.

Цель статьи – описание алгоритма синтеза тестов для ОС без

наблюдаемых выходов $\hat{Y}$ и анализ сложности процедуры синтеза.

Синтез проверяющих последовательностей по тестовому графу ячейки сети. В основе построения тестового графа и его использования для нахождения множества проверяющих тестов лежит понятие различимости состояний ячеек сети. Говорят, что два состояния ячейки сети z_i и z_j различимы, если существует по меньшей мере один входной вектор x, приложение которого вызывает появление на наблюдаемых выходах различных реакций в том случае, когда z_i и z_j приложены к левому входу первой ячейки сети. Множество пар различимых состояний ТПВ ячейки ОС можно найти, построив таблицу переходов пар состояний или эквивалентный граф.

Тестовый граф одномерной однородной сети определяется, как граф $G = (V, E)$, у которого множество вершин V равно:

$$V = \{(z_i, z_j) I z_i \neq z_j\} \text{ и } \{z_k I \delta(z_i, x_\alpha) = \delta(z_j, x_\alpha) = z_k, z_i \neq z_j\},$$

где $\delta(z_i, x_\alpha), \delta(z_j, x_\alpha)$ определяется из таблицы переходов ячейки сети.

Из вершины (z_i, z_j) выходит дуга $e \in E$, входящая в вершину (z_a, z_a), если имеется некоторый входной символ x_α такой, что

1) $\delta(z_i, x_\alpha) = z_a$, $\delta(z_j, x_\alpha) = z_в$;

2) $\delta(z_a, x_\alpha) = z_j$, $\delta(z_в, x_\alpha) = z_i$

где z_a не обязательно отличается от z_a, а (z_i, z_j) не обязательно отличается от (z_a, z_a). Каждая дуга отмечается вход/выходной парой $x_\alpha / \hat{Y}_\alpha$, а в сети без наблюдаемого выхода $\hat{Y}$ - входным символом x_α Кроме того, если выполняются переходы в соответствии с вариантом 2, то дуга соединяющая вершины (z_i, z_j), (z_a, z_a) тестового графа отмечается дополнительно крестиком.

Как было показано в [8], любая вершина в тестовом графе, имеющая петлю, представляет пару различимых состояний. Эти состояния различаются на выходе $\hat{z}$ крайней правой ячейки сети при приложении к входам x двоичного набора, определяемого весом петли. Аналогично, если существует замкнутый цикл в тестовом графе, то всегда можно найти входной набор x, соответствующий этому циклу, приложение которого обеспечивает различимость пар состояний, входящих в этот замкнутый цикл. Если в тестовом графе существует путь из некоторой вершины $V_i = (z_a, z_в)$ к вершине V_j, входящей в некоторый замкнутый цикл, то пара состояний

(z_a, z_a) является также различимой.

Процедуру синтеза проверяющих тестов для одномерных ОС основанную на использовании тестового графа ячейки сети, рассмотрим на примере одномерной ОС без наблюдаемых выходов x, представленную на рис. 2. Сеть состоит из шести однотипных ячеек. Функциональная схема ячейки сети (рис. 3) определяет таблицу переходов автоматной модели ячейки, которая представлена в таблице 1 или в упрощенном виде в таблице 2.

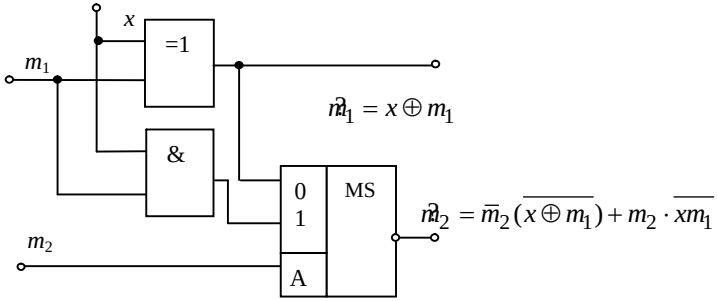

Рис. 3. Функциональная схема ячейки ОС

Таблица 1 – Кодированная таблица переходов ячейки ОС

m_1, m_2	m_1', m_2'	
	$x = 0$	$x = 1$
00	01	10
01	01	11
10	10	01
11	11	00

Таблица 2 – Таблица переходов ячейки ОС

$z(t)$	$z(t+1)$	
	$x_0 = 0$	$x_1 = 1$
z_0	z_1	z_2
z_1	z_1	z_3
z_2	z_2	z_1
z_3	z_3	z_0

Тестовый граф для рассматриваемой ячейки, построенный по ее таблице переходов, представлен на рис. 4 Он состоит из двух связных компонент, в каждой из которых можно выделить множество фундаментальных циклов, то есть таких циклов, в которых имеется, по меньшей мере, одна дуга, не принадлежащая никакому другому циклу.

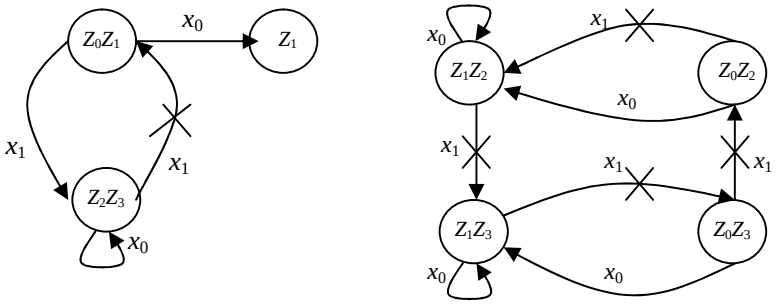

Рис. 4. Тестовый граф ячейки

Множество фундаментальных циклов тестового графа рассматриваемой сети представлено на рис. 5. Если тестовый граф содержит фундаментальный цикл, который в общем виде показан на рис. 6,

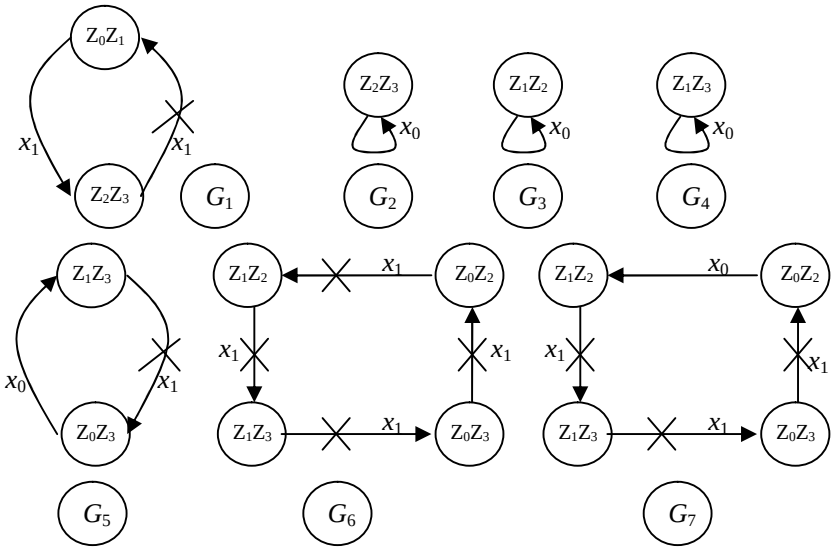

Рис. 5. Множество фундаментальных циклов тестового графа

то для обнаружения всех неисправностей типа $(z_i \to z_j)$, $(z_j \to z_i)$, $(z_a \to z_a)$ и $(z_a \to z_a)$ в любой ячейке сети достаточно приложить следующие четыре теста:

1) $z_i \overset{x_\alpha}{\text{—}} z_a \overset{x_\beta}{\text{—}} z_i \overset{x_\alpha}{\text{—}} z_a \overset{x_\beta}{\text{—}} \ldots 3)$ $z_j \overset{x_\alpha}{\text{—}} z_a \overset{x_\beta}{\text{—}} z_j \overset{x_\alpha}{\text{—}} z_a \overset{x_\beta}{\text{—}} \ldots$

2) $z_a \overset{x_\beta}{\text{—}} z_i \overset{x_\alpha}{\text{—}} z_a \overset{x_\beta}{\text{—}} z_i \overset{x_\alpha}{\text{—}} \ldots 4)$ $z_a \overset{x_\beta}{\text{—}} z_j \overset{x_\alpha}{\text{—}} z_a \overset{x_\beta}{\text{—}} z_j \overset{x_\alpha}{\text{—}} \ldots$

Независимо от числа ячеек сети эти тесты вызывают появление различных состояний z в одной и той же ячейке, а, следовательно, перечисленные выше неисправности обнаруживаются на наблюдаемом выходе z_2 сети. Так как множество переходов тестового графа рассмотренной выше сети покрывается множеством фундаментальных циклов рис. 5, за исключением перехода $(z_0 z_1) \overset{x_0}{\text{—}} z_1$, то для каждого цикла можно определить множество тестов, проверяющих правильность переходов и состояний каждой ячейки сети. На рис. 7 для каждого фундаментального цикла $G_1 - G_7$ тестового графа ячейки сети приведены тесты, позволяющие обнаружить неисправную ячейку сети.

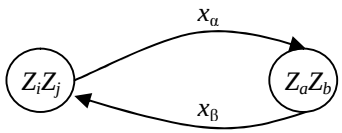

Рис. 6. Цикл в тестовом графе

Однако в тестовом графе имеется переход $(z_0 z_1) \overset{x_0}{\text{—}} z_1$, который не входит ни в один из рассмотренных выше циклов и порождает неопределенность при проверке состояний ячеек, сети тестами $t_7, t_{11}, \ldots t_{12}$ е t_{13}. Для исключения этой неопределенности тест t_7 необходимо расширить тестам $t_{22} \div t_{26}$ тесты t_{11}, t_{12}, t_{13} - тестами t_{27}, t_{29}, t_{31} соответственно. Рассмотренная выше методика синтеза проверяющих тестов, основанная на использовании тестового графа ячейки сети и выделении множества фундаментальных циклов в тестовом графе, имеет следующие особенности.

Во-первых, сложность процедуры синтеза определяется сложностью построения тестового графа и процедуры нахождения фундаментальных циклов в графе.

Во-вторых, тесты, построенные по фундаментальным циклам тестового графа ячейки сети, являются, как правило, избыточными. Например, тест, определяемый из цикл, а G_3 на рис. 5. $t_8 : z_2 \overset{x_0}{\text{—}} z_2 \overset{x_0}{\text{—}} z_2 \ldots$ эквивалентен тесту

t_5, определяемому из цикла G_2. Как видно из рис. 7, множество тестов, получаемых из фундаментальных циклов тестового графа, для рассматриваемой сети содержит 13 эквивалентных тестов.

Цикл	Тесты	Цикл	Тесты
G_1	$t_1 : z_0^{x_1} z_2^{x_1} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1$	G_7	$t_{18} \sim t_{11}$;
	$t_2 : z_2^{x_1} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1^{x_1} z_3$		$t_{19} \sim t_{12}$;
	$t_3 : z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1^{x_1} z_3^{x_1} z_0$		$t_{20} \sim t_{13}$;
	$t_4 : z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2$		$t_{21} \sim t_5$
G_2	$t_5 : z_2^{x_0} z_2^{x_0} z_2^{x_0} z_2^{x_0} z_2^{x_0} z_2^{x_0} z_2$	t_7	$t_{22} : z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_1} z_3$
	$t_6 : z_3^{x_0} z_3^{x_0} z_3^{x_0} z_3^{x_0} z_3^{x_0} z_3^{x_0} z_3$		$t_{23} : z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_1} z_3^{x_1} z_0$
			$t_{24} : z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2$
G_3	$t_7 : z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1^{x_0} z_1$		$t_{25} : z_1^{x_0} z_1^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1$
	$t_8 : t_8 \sim t_5$		$t_{26} : z_1^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1^{x_1} z_3$
G_4	$t_9 : t_9 \sim t_7$,	t_{11}	$t_{27} : z_1^{x_1} z_3^{x_1} z_0^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2$
	$t_{10} : t_{10} \sim t_6$		$t_{28} : t_{28} \sim t_3$
G_5	$t_{11} : z_1^{x_1} z_3^{x_1} z_0^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_0} z_1$	t_{12}	$t_{29} : z_3^{x_1} z_0^{x_1} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1$
	$t_{12} : z_3^{x_1} z_0^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_0} z_1^{x_1} z$		$t_{30} : t_{30} \sim t_4$
	$t_{13} : z_0^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_0} z_1^{x_1} z_3^{x_1} z_0$		
G_6	$t_{14} \sim t_1$;	t_{13}	$t_{31} : z_0^{x_0} z_1^{x_1} z_3^{x_1} z_0^{x_1} z_2^{x_1} z_1^{x_1} z_3$
	$t_{15} \sim t_2$;		
	$t_{16} \sim t_3$;		
	$t_{17} \sim t_4$		

Рис. 7. Множество проверяющих тестов одномерной ОС рис. 2, рис. 3

В-третьих, полученные тесты могут содержать множество неопределенных переходов, появление которых обусловлено наличием в ТПВ

ячейки пар совместимых состояний. Поэтому, как было указано выше, полученные тесты необходимо анализировать с целью нахождения таких неопределенных переходов и последующего расширения множества проверяющих тестов.

Оценка сложности процедуры синтеза проверяющих тестов по тестовому графу ОС. Если n- число состояний ячейки сети, то тестовый граф содержит число вершин $g = 0(n^2)$. Число фундаментальных циклов в графе находится путем добавления к этому остову произвольной дуги, не принадлежащей остову графа, и проверке - является ли циклом контур из остовных дуг графа и добавленной дуги? Если в графе имеется t дуг, (g-1) из которых принадлежат остову, то число всех фундаментальных циклов, построенных таким способом, равно цикломатическому числу [9].

$$U = t - g + b,$$

где b- число связных компонент тестового графа, $b \approx 0(n)$.

Так как $t \geq g \times g \approx (n^4)$, то эта оценка является верхней границей трудоемкости рассмотренной выше процедуры синтеза проверяющих тестов. Следует отметить, что в полученной оценке не учитывается сложность нахождения остовного подграфа тестового графа ячейки сети. Процедура анализа избыточности тестов, проверяющих исправность ОС, и их расширения по трудоемкости эквивалентна процедуре моделирования неисправностей на функциональном уровне описания ячеек сети.

Выводы. В статье представлена процедура синтеза проверяющих последовательностей по тестовому графу ячейки ОС без наблюдаемых выходов $\mathcal{Y}$, которая является развитием подхода, предложенного в [8]. Получена верхняя граница трудоемкости процедуры синтеза тестов.

Список литературы: **1**. *Тофоли Т., Марголус Н.* Машины клеточных автоматов. –М.: Из-во «Мир», – 1991. – 280с., **2.** Варшавский В.И., Мараховский В.Б. Однородные структуры. Анализ. Синтез. Поведение. –М.: Энергия, –1973. –152с., **3**. *Визирев И.С., Гузик В.Ф. и др. Синтез управляющих устройств в однородных средам.* –М.: Наука. –1984. –166с., **4.** *Евренков Э.В., Прангишвили И.В.* Цифровые автоматы с настраиваемой структурой. –М.: Энергия, –1974. –240с., **5.***Чараев В.Г.* Контроль исправности и диагностика неисправностей однородной двумерной структуры // Автоматика и телемеханика. –1968. –№7. с.45-52., **6.** *Cheng W.T., Patel J.N.* Multiple fault-detection in iterative logic arrays. – P.493-499., **7.** Kautz W.H. Testing for faults in cellular logic arrays. – Proc. 8-th Annuel Symp. Switching and Automata Theory. – 1967. – p.161-174., **8.** Friedman A.D., Menon P.R. Fault detection in digital circuits. – New Jersey: Prentice Hall. – 1971. – 220p. **9.** *Кристофидис Н*. Теория графов. Алгоритмический поход. – М.: Мир, 1978. – 432с.

Поступила в редколлегию 16.04.08

УДК 681.3

Л.В. ДЕРБУНОВИЧ, д-р. техн. наук, проф. каф. АУТС НТУ "ХПИ",
В.С. СУЗДАЛЬ, д-р. техн. наук, ***Ю.М. ЕПИФАНОВ***, канд. техн. наук,
Л.И. ГЕРАСИМЧУК, канд. техн. наук (Институтт сцинтилляционных материалов НАН Украины), ***Ю.С. КОЗЬМИН***, инженер ИСМА

СИСТЕМА УПРАВЛЕНИЯ ВЫРАЩИВАНИЕМ СЦИНТИЛЛЯЦИОННЫХ МОНОКРИСТАЛЛОВ

В статті розглянуто систему управління багатомірним технологічним процесом вирощування методом Чохральського сцинтиляційних монокристалів великих розмірів; визначені задачі оптимального управління режиму установок типу "РОСТ"; застосування розроблених методів і засобів дозволило підвищити вихід придатної продукції.

In this paper viewing algorithms of MIMO-system control for Czochralski technological process growing of large single crystals; was define problems of control and searching optimal heat conditions for growth plant type “ROST”; carrying out of useful productions was increased by using this methods and tools.

Постановка проблемы. Сцинтилляционные монокристаллы (СМК) выращивают из расплава методом Чохральского, который заключается в вытягивании монокристалла из расплава, находящегося в тигле при температуре выше точки плавления сырья, на затравку, прикрепленную к охлаждающему стержню - кристаллодержателю [1]. В основе метода Чохральского, как и других методов выращивания монокристаллов из расплава, заложены принципы направленной кристаллизации, которая осуществляется при наличии и взаимодействии двух направленных потоков – переноса тепловой энергии и межфазного массопереноса. В методе Чохральского боковая поверхность образца формируется без контакта со стенками тигля, поэтому форма и размер кристалла определяются капиллярными силами, формирующими мениск расплава, и условиями тепло - массообмена в системе кристалл - расплав.

Форма выращиваемого кристалла в основной его части является цилиндрической, а его качество (степень бездефектности и однородность распределения примеси) в значительной степени определяется стабильностью геометрии поверхности образца. К параметрам, оказывающим существенное влияние на условия кристаллизации СМК в установках "РОСТ", относятся: температура в ростовой камере, уровень расплава в тигле, скорость вытягивания кристалла. С целью стабилизации диаметра кристалла в существующих SISO-системах осуществляется замкнутое управление тепловым режимом кристаллизации, которое обеспечивает точность стабилизации диаметра растущего образца ~3?5% [2-4].

Технологический процесс выращивания СМК относится к