УДК 681.32

СИНТЕЗ ПРОВЕРЯЮЩИХ ТЕСТОВ ДЛЯ ОДНОРОДНЫХ СХЕМ

Г.Г. Четвериков[1], Я.Ю. Королева[2], М.А. Бережная[3]

[1] ХНУРЭ, г. Харьков, Украина, chetvergg@kture.kharkov.ua;
[2] ХНУРЭ, г. Харьков, Украина;
[3] ХНУРЭ, г. Харьков, Украина

Проведен анализ методов тестового диагностирования однородной структуры (ОС), определены множества отличительных и характеристических символов автоматных диаграмм ОС. Предложен способ их нахождения по характеристическому дереву – модели ячейки сети, определен класс циклических отличительных последовательностей ОС и необходимые условия существования в данной ОС циклических отличительных последовательностей.

ОДНОРОДНАЯ СТРУКТУРА, ТЕСТОВОЕ ДИАГНОСТИРОВАНИЕ, ОТЛИЧИТЕЛЬНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

Введение

Микропроцессорные СБИС, программируемые логические интегральные схемы (ПЛИС) типа FPGA и CPLD широко используются в разработках современных вычислительных и управляющих устройств, являются топологической основой для создания высокопроизводительных однородных структур или сетей (ОС) систолической обработки информации. Существует ряд работ, рассматривающих различные аспекты проблемы синтеза ОС по заданному алгоритму обработки информации [1-4]. Отличительной чертой такой обработки является однотипность структуры ячеек-модулей сети, выполняемых ею операций, связей между ячейками, рекурсивность вычислений.

Второй проблемой, возникающей при синтезе ОС, является сложность процедур верификации и тестового диагностирования сети [4-6]. На рис. 1 представлена структура одномерной ОС, в которой левый вход $z(1)$ и все вертикальные выходы $x(i)$, $i=\overline{1,p}$ являются управляемыми, а выходы $\hat{x}(i)$ и $\hat{z}_p$ – наблюдаемыми.

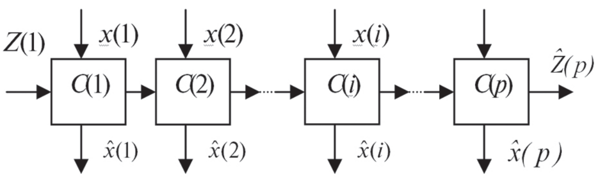

Рис. 1. Одномерная ОС с наблюдаемыми выходами $\hat{x}$

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

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.

Проблема тестового диагностирования одномерных ОС была исследована в работе [7], в которой были предложены методы тестирования неисправностей класса F_1 в ОС. В [8] определены необходимые и достаточные условия 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.

Недостатки существующих методов синтеза проверяющих тестов для ОС без наблюдаемых выходов $\hat{x}$ можно исключить, если воспользоваться подходом, основанном на использовании характерных последовательностей автоматной модели ячейки сети и построении проверяющих тестов на основе циклических отличительных последовательностей ячейки ОС.

1. Отличительные последовательности в ОС

На функциональном уровне описания ячейки одномерной ОС без наблюдаемых выходов $\hat{x}$ будем рассматривать ее таблицу истинности как таблицу переходов-выходов (ТПВ) автомата Мура, задаваемого тройкой (X, Z, δ), у которого функции переходов и выходов совпадают, то есть $\delta(z_i, x_a) = \lambda(z_i, x_a) = \hat{z}_i$, $\forall z_i, z_j \in Z$, $\forall x_a \in X$.

Пусть $Z = \{z_1, z_2, \ldots, z_i, \ldots, z_n\}$ – множество состояний автоматной модели ячейки сети. Будем обозначать $Z / z_i = \{z_1, z_2, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n\} = \{Z - z_i\}$ – множество состояний ячейки сети, из которого исключен элемент z_i. Как было показано в [8, 9], при синтезе проверяющих тестов сложность решения задачи анализа полноты полученных тестов и их расширения связана с наличием в автоматной модели ячейки сети пар x_α-совместимых состояний.

Определение 1. Если столбец x_α таблицы переходов ячейки сети содержит пару состояний (z_a, z_b), для которых состояния-преемники эквивалентны, то есть $\delta(z_a, x_\alpha) = z_k$ и $\delta(z_b, x_\alpha) = z_k$, то состояния z_a и z_b являются x_α совместимыми или неразличимыми для входного символа x_α.

В общем случае может быть две альтернативы. Пара состояний (z_a, z_b) различается по меньшей мере одним входным символом x_β, либо пара (z_a, z_b) – x_i-совместима для каждого входного символа $x_i \in X$. В первом случае состояния z_a и z_b можно идентифицировать только приложением ко входу проверяемой ячейки входного символа x_β.

Из табл. 1 переходов ячейки легко можно определить множества $z_c^{x_i} x_i$ – совместимых состояний для каждого входного символа $x_i \in X$, $i = \overline{1, m}$. Образуем множество $Z_c = \{Z_c^{x_1} \cap Z_c^{x_2} \cap \ldots \cap Z_c^{x_m}\}$, элементы которого представляют собой состояния, x_α-совместимые для всех входных символов множества X.

Таблица 1

Таблица переходов ячейки

$z(t)$	$z(t+1)$		
	x_1	x_2	x_3
z_1	z_1	z_2	z_2
z_2	z_1	z_3	z_2
z_3	z_1	z_4	z_4
z_4	z_4	z_4	z_2

Определение 2. Пусть ячейка одномерной ОС без наблюдаемых выходов $\hat{x}$ представлена моделью автомата Мура (X, Z, δ), в котором x_α-преемником состояния $\forall z_i \in Z$ является состояние z_k, не обязательно отличное от z_i. Входной символ $x_\alpha \in X$ будем называть отличительным символом состояние z_i тогда и только тогда, когда z_k не является x_α-преемником для множества Z / z_i начальных состояний автомата и $z_k \notin z_c$.

Для различения состояний (z_a, z_b) можно воспользоваться множеством характеристических входных символов подобно тому, как в [10] использовались характеристические последовательности при построении диагностических экспериментов для автоматов, не имеющих отличительных последовательностей.

Определение 3. Пусть ячейка одномерной сети представлена моделью автомата Мура (X, Z, δ). Множество входных символов $X_c = \{x_1, x_2, \ldots x_r\}$ называется множеством входных характеристических символов тогда и только тогда, когда для любой пары состояний $(z_a, z_b) \in Z$ автомата $\delta(z_a, x_1)\delta(z_a, x_2)\ \ldots\ \delta(z_a, x_r) \neq \delta(z_b, x_1)\delta(z_b, x_2)\ \ldots\ \delta(z_b, x_r)$.

Множество отличительных и характеристических символов может быть найдено из характеристического дерева автомата ячейки сети, которое строится по правилам, приведенным в [10].

Характеристическое дерево ячейки сети, представленной автоматной диаграммой в табл. 1, приведено на рис. 2, из которого можно найти множество отличительных и характеристических символов.

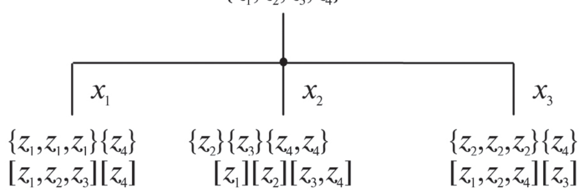

Рис. 2. Характеристическое дерево ячейки

Так как произведение разбиений

$$\pi(x_1) = (\overline{z_1 z_2 z_3}, \bar{z}_4) \text{ и } \pi(x_2) = (\bar{z}_1, \bar{z}_2, \overline{z_3 z_4})$$

равно $\pi(0)$, то множество $X_c = \{x_1, x_2\}$ является множеством входных характеристических символов. Простые σ-множества для каждой вершины характеристического дерева определяют множество состояний, для которых метка вершины дерева является отличительным входным символом. Исключения составляют лишь те простые σ-множества, которые являются состояниями преемниками, принадлежащими множеству Z_c, то есть множеству состояний, неразличимых для всех входных символов.

Например из характеристического дерева рис. 2 ячейки сети (табл. 1) следует, что $z_c^{x_1} = \{z_1, z_2, z_3\}$, $z_c^{x_2} = \{z_3, z_4\}$, $z_c^{x_3} = \{z_1, z_2, z_4\}$. Множества x_α-совместимых состояний ячейки позволяют определить множество Z_c в виде: $Z_c = Z_c^{x_1} \cap Z_c^{x_2} \cap Z_c^{x_3} = \varnothing$.

Определив множество Z_c, из характеристического дерева можно легко найти множество отличительных символов и состояний ячейки, которые различаются этими символами в соответствии с определением 2. Для рассматриваемого примера состояние z_4 различается символом E_1, состояния z_1 и z_2 – символом E_2, состояние z_3 – символом E_3.

Определение 4. Пусть состояние z_j правого выхода ячейки $C(k)$ одномерной ОС транспортируется на крайний правый выход сети приложением входного набора X_T ко входам ячеек $C(k+1), C(k+2), \ldots, C(p)$. Входной набор X_T является проверяющим тестом, идентифицирующим состояние z_j ячейки $C(k)$, если состояние наблюдаемых выходов сети различно, когда z_j и каждое из состояний множества Z / z_j приложено к левому входу ячейки $C(k+1)$, то есть $\delta(z_j, X_T) \neq \delta(z_i, X_T)$, $\forall z_i \in Z / z_j$.

Если для каждого состояния ячейки сети существует проверяющий входной набор X_T, то проверяющий тест всей ОС можно построить путем проверки правильности всех $m \cdot n$ переходов автоматной модели каждой ячейки сети, используя проверяющие тестовые наборы X_T для идентификации правильности переходов.

Нижеследующая теорема определяет необходимые и достаточные условия существования в однородной сети проверяющих тестовых наборов X_T.

Теорема 1. Пусть правый выход ячейки $C(k)$ одномерной однородной сети находится в состоянии z_j, и к верхним входам ячеек $C(k+1), C(k+2), \ldots, C(p)$ приложен входной набор $X_k = (E_{k+1}, E_{k+2}, \ldots, E_p)$, вызывающий последовательность переходов состояний ячеек сети в виде $z_j \xrightarrow{E_{k+1}} z_{k+1} \xrightarrow{E_{k+2}} z_{k+2} \xrightarrow{E_{k+3}} \ldots$ $z_{p-1} \xrightarrow{E_{p-1}} z_p$, где состояние слева от входного символа $E_j \in X_k$, $\alpha = (k+1)(k+2)\ldots$ является предшествующим, а состояние справа от x_α – последующим. Если существует последовательность состояний ячеек $C(k+1), C(k+2), \ldots, C(p)$, порождаемая приложением входного набора X_k, в котором каждый символ $E_\alpha \in X_k$ является отличительным символом предшествующего состояния, то последовательность этих символов образует проверяющий тестовый набор состояния z_j ячейки $C(k)$ сети.

Доказательство. Необходимость. По условию теоремы символ x_{k+1} является отличительным для состояния z_j, x_{k+2} – для z_{j+1} и так далее. Пусть в результате воздействия неисправности на правом выходе ячейки $C(k)$ вместо состояния z_j появилось состояние z_i. Так как X_{k+1} является отличительным символом состояния z_j, то в соответствии с определением 2 для каждого $z_i \in Z / z_i$. Следовательно, при наличии неисправности $z_j \to z_i$ на правом выходе ячейки $C(k+1)$ появится состояние $z_{k+1}^* \neq z_{k+1}$. В свою очередь, так как x_{k+2} – отличительный символ состояния z_{k+1}, то $\delta(z_{k+1}, x_{k+2}) \neq \delta(z_{k+1}^*, x_{k+2}^*)$ и так далее для каждой ячейки сети. В результате на наблюдаемых выходах ячейки $C(p)$ имеем $\delta(z_{p-1}^*, x_p) \neq \delta(z_{p-1}, x_p)$. Следовательно, состояние наблюдаемого выхода сети отличает состояние z_j ячейки $C(k)$ от любого другого состояния, и в соответствии с определением 4 входной набор $z_i \in Z / z_i$ является тестовым набором, обнаруживающим неисправность типа $z_j \to Z / z_j$ в ячейке $C(k)$.

Достаточность. Пусть входной набор x_k, приложенный ко входам ячеек $C(k+1), C(k+2), \ldots, C(p)$ сети является проверяющим тестовым набором, обнаруживающим любую неисправность ячейки $C(k)$ вида $z_j \to Z / z_j$. Предположим, что входной символ x_{k+1}, приложенный ко входу ячейки $C(k+1)$, не является отличительным символом предшествующего состояния z_j ячейки $C(k)$. В соответствии с определением 3 существует по меньшей мере одно состояние z_a, которое совместимо с z_j, то есть $\delta(z_j, x_{k+1}) = \delta(z_a, x_{k+1}) = z_{k+1}$. Поэтому неисправность типа $z_j \to z_i$ не будет транспортироваться на наблюдаемый выход сети. Следовательно, символ X_{k+1} должен быть отличительным символом состояния z_j. Рассуждая аналогично, нетрудно убедиться в том, что этому условию должен удовлетворять каждый символ $x_\alpha \notin x_k$.

2. Синтез тестов на основе циклических отличительных последовательностей

Задача построения проверяющих входных наборов для заданной одномерной ОС значительно упрощается, если существуют циклические переводящие последовательности, образованные из отличительных входных символов. При построении диагностических экспериментов по автоматным моделям дискретных устройств (ДУ) возникает задача нахождения множества переводящих последовательностей $T(z_i, z_j)$, с помощью которых ДУ из начального состояния z_i переводится в состояние z_j. Известно, что эта задача тривиальна, если ДУ задано графом или таблицей переходов [10], переводящая последовательность $T(z_i, z_j)$ определяется путем построения фрагментов дерева преемников состояния z_i, содержащего путь или множество путей, оканчивающихся состоянием z_j.

При построении проверяющих тестов для одномерных ОС возникает аналогичная задача нахождения множества переводящих последовательностей по автоматной модели ячейки сети. Тестопригодность сети, число ячеек которой превышает число состояний автоматной модели ячейки сети, определяется наличием в ней циклических переводящих последовательностей $T(z_i, z_j)$, которые так же, как и любые другие переводящие последовательности, можно легко найти из автоматной модели ячейки сети.

Определение 5. Циклическую переводящую последовательность, образованную из входных отличительных символов предшествующих состояний ячеек сети, будем называть циклической отличительной последовательностью (ЦОП).

В соответствии с теоремой 1 циклическая переводящая последовательность $T(z_i, z_j)$, которая образована из отличительных входных символов, является одновременно проверяющим тестовым набором, который транспортирует состояние проверяемой ячейки сети на наблюдаемый выход.

Рассмотрим класс однородных одномерных сетей без наблюдаемых выходов $\hat{x}$, состоящих из p ячеек, каждая из которых имеет m входных символов, n состояний и каждое состояние имеет по меньшей мере один отличительный символ. Для сетей такого типа справедливо следующее утверждение.

Теорема 2. Если в ячейке ОС с n состояниями каждое состояние имеет по крайней мере один отличительный символ, то для такой сети существует по меньшей мере одна циклическая отличительная последовательность.

Доказательство. Циклическая последовательность переходов состояний автомата отсутствует, когда состояния-преемники не совпадают с состояниями-предшественниками, то есть когда выполняется условие $\delta(\underline{z_i}, x_i) = z_j$, $z_i \neq z_j$, $\forall x_i \in X(i = \overline{1,m})$, $\forall z_i, z_j \in Z(i, j = \overline{1,n})$. Не нарушая общности рассуждений, представим последовательность переходов состояний, удовлетворяющую этому условию, в следующем виде:

$$z_1 \frac{E_1}{\quad} z_2 \frac{E_2}{\quad} z_3 \;\ldots\; z_i \frac{E_i}{\quad} z_j \;\ldots\; \frac{E_{n-1}}{\quad} z_n .$$

В этой последовательности нет циклов. Однако по условию теоремы каждое состояние имеет отличительный символ, в том числе и последнее состояние цепи z_n. Пусть x_n – отличительный символ z_n. Так как рассмотренная цепь включает все n состояний, то $\delta(z_n, x_n) = z_i$, где индекс «i» может принимать любое значение от 1 до n. Следовательно, существует переход в одно из предшествующих вершин цепи, а значит, существует по меньшей мере один цикл, образованный отличительными символами.

В качестве примера рассмотрим автомат, заданный табл. 1. Из характеристического дерева автомата (рис. 2) находим отличительные символы состояний: $z_1 - z_2$; $z_2 - z_2$; $z_3 - z_3$; $z_4 - z_1$.

Теорема 2 определяет необходимые условия существования в ячейке ОС циклической отличительной последовательности. Существование ЦОП для заданной однородной сети упрощает процедуру построения проверяющего эксперимента для этой сети так же, как и наличие отличительных последовательностей для автоматных моделей произвольных ДУ с элементами памяти. Такие ОС являются легко тестируемыми.

Выводы

На основе анализа методов тестового диагностирования ОС предложен единый методологический подход к решению этой проблемы, основанный на функциональном принципе описания поведения ОС, функциональных моделях неисправностей и использовании методов теории экспериментов с автоматами. Определены множества отличительных и характеристических символов автоматных диаграмм ОС, предложен способ их нахождения по характеристическому дереву – модели ячейки сети. ОС, ячейка которой для каждого состояния имеет отличительный символ, является легко тестируемой. Определены необходимые и достаточные условия существования в однородной сети проверяющих тестовых наборов, которые зависят от свойств автоматной модели ячейки сети и существовании последовательностей состояний, имеющих отличительные символы. Определен класс циклических отличительных последовательностей ОС и необходимые условия существования в данной ОС циклических отличительных последовательностей, формируемых из отличительных символов предшествующих состояний ячеек сети, которые позволяют формализовать и упростить процедуру и построения проверяющего эксперимента ОС.

Список литературы: 1. Тоффоли Т. Машины клеточных автоматов [Текст] / Т. Тоффоли, Н. Марголус. – М.: Мир, 1991. – 280 с. 2. Варшавский В.И. Однородные структуры. Анализ. Синтез. Поведение. [Текст] / В.И. Варшавский, В.Б. Мараховский. –М.: Энергия, 1973. – 152 с. 3. Синтез управляющих устройств в однородных средам [Текст] / И.С. Визирев,. В.Ф Гузик и др. – М.: Наука, 1984. – 166 с. 4. Евренков Э.В. Цифровые автоматы с настраиваемой структурой [Текст] / Э.В. Евренков, И.В. Прангишвили. – М.: Энергия, 1974. – 240 с. 5. Чараев В.Г. Контроль исправности и диагностика неисправностей однородной двумерной структуры [Текст] / В.Г. Чараев // Автоматика и телемеханика. – 1968. – №7. – С. 45-52. 6. Cheng W.T. Multiple fault-detection in iterative logic arrays / W.T. Cheng, J.N. Patel // Proc. 1985 Int. Test Conference. – Nov., 1985. – P.493-499. 7. Kautz, W.H. Testing for faults in cellular logic arrays / W.H. Kautz // Proc. 8-th Annuel Symp. Switching and Automata Theory. – 1967. – P. 161-174. 8. Friedman, A.D. Fault detection in digital circuits / A.D. Friedman, P.R. Menon – New Jersey: Prentice Hall, 1971. – P.220. 9. Фридман А. Теория и проектирование переключательных схем [Текст] / А. Фридман, Д. Менон. – М.: Мир, 1978. – 589 с. 10. Тоценко В.Г. Алгоритмы технического диагностирования дискретных устройств [Текст] / В.Г. Тоценко. – М.: Радио и связь, 1983. – 240 с.

Поступила в редколлегию 14.10.2009

УДК 681.32

Синтез комбінаційних сем у базисі поліноміальних форм / Г.Г. Четверіков, Я.Ю. Корольова, М.А. Бережна // Біоніка інтелекту: наук.-техн. журнал. – 2009. – № 2 (71). – С. 123-127.

Представлено метод синтезу перевіряючих тестів для класу одномірних ОС, у яких відсутні виходи $\hat{x}$, оснований на використанні автоматних моделей чарунок ОС і функціональних моделях пошкоджень чарунок мережі. Запропоновано і обгрунтовано використання циклічних і відрізняючих послідовностей чарунки мережі для побудови діагностичного експеримента в ОС.

Іл. 2. Бібліогр.: 10 найм.

UDK 681.32

Synthesis of combinational circuits in polynomial forms BASIS / G.G. Chetverikov, Y.U. Koroleva, M.A. Beregnaya // Bionics of Intelligence: Sci. Mag. – 2009. – № 2 (71). – P. 123-127.

The test method of one-dimensional 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 cyclic distinguishing sequences of automation model of the ILA cell.

Fig. 2. Ref.: 10 items.