

4. Мирошник М.А. Методы параллельного решения SAT-задач для реализации процедур прогнозирования трудоемкости. / Мирошник М.А., Клименко Л.А., Герман Э.Е. // Інформаційно-керуючі системи на залізничному транспорті: науково-технічний журнал. №3 - 2016 –с. 23-29.

5. Мирошник М.А. Методы автоматизации проектирования легкотестируемых компьютерных систем и устройств на основе цифровых автоматов. / Мирошник М.А., Клименко Л.А., Пахомов Ю.В. // Інформаційно-керуючі системи на залізничному транспорті: науково-технічний журнал. №4. – 2018. - С. 3-10.

*Мирошник М. А., д.т.н., профессор (УкрГУЖТ),
Королева Я. Ю., доцент, (НТУ «ХПИ»)*

УДК 681.3

СИНТЕЗ ЛЕГКОТЕСТИРУЕМЫХ ДВУМЕРНЫХ СЕТЕЙ

В докладе проведен анализ методов тестового диагностирования двумерных однородных сетей (ОС), предложен метод модификации автоматной модели сети, обеспечивающий С - тестируемость строк и L – тестируемость столбцов сети [1].

За последние десятилетия появилось множество работ, посвященных использованию клеточных сетей из однородных модулей для реализации параллельных и распределенных вычислений, моделирования сложных динамических систем, блочного и поточного шифрования в криптосистемах. Однородные сети представляют собой упорядоченный массив однородных клеточных автоматов в n-мерном пространстве, в котором каждая клетка имеет ограниченное множество состояний (от 2 до 32) и ограниченными связями с ближайшими соседями. В соответствии с набором правил настройки каждая клетка вычисляет свое новое состояние на каждом такте функционирования сети, что обеспечивает сколь угодно большую степень параллелизма. Двумерная структура однородной сети идентична структуре матричной ПЛИС типа FPGA, в которой каждый конфигурируемый логический блок или некоторый набор блоков программируется в качестве клеточного автомата однородной сети [2].

В работах зарубежных авторов развиваются методы тестового диагностирования, основанные на функциональном подходе и использовании автоматных моделей ячеек сети. 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) [5].

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 функциональных свойств КА 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.

Анализ перечисленных выше работ в области тестового диагностирования однородных сетей показывает, что, несмотря на имеющиеся достижения и большое число работ в этой области, отсутствует единый методологический подход к решению этой проблемы для двумерных однородных сетей различного класса.

В докладе представлена разработка единого методологического подхода к решению задачи тестового диагностирования двумерных однородных сетей, основанного на использовании автоматных моделей клеточных автоматов теории экспериментов с автоматами и разработка метода модификации автоматной диаграммы клеточных автоматов, обеспечивающей С-тестируемость двумерных однородных сетей для класса неисправностей F_1 .

Анализ существующих методов тестового диагностирования двумерных однородных сетей и их преобразований к тестопригодному виду показывает, что использование необходимых и достаточных условий построения С-тестируемых сетей в большинстве случаев ограничивается сложностью аппаратной реализации этих условий. С целью сокращения аппаратных затрат предложено совмещать реализацию С- тестируемых строк с L-тестируемостью столбцов двумерной матрицы. Предложенная структура ячейки сети, обладающая указанными выше свойствами.

Список литературы

1. Miroshnik M.A. Implementation of cryptographic algorithms on FPGA-based digital distributed systems. / Miroshnik M.A. // Інформаційно-керуючі системи на залізничному транспорті: науково-технічний журнал. № 2 (111). – 2015. – С. 25-30.
2. Мирошник М.А. Методы защиты информации в распределенных компьютерных сетях. / Мирошник М.А. // Інформаційно-керуючі системи на залізничному транспорті: науково-технічний журнал. №5. – 2014. – с. 66-70.
3. Miroshnik M.A. Design of a Built-in Diagnostic Infrastructure for Fault-Tolerant Telecommunication Systems. / Miroshnik M.A. // Modern problems of Radio Engineering, Telecommunications and Computer Science - Proceedings of the 11th International Conference, TCSET'2012.
4. Мирошник М. А. Синтез распределенных вычислительных сред на базе компьютерных сетей. / Мирошник М.А. // Системи обробки інформації. - №7 - 2013 –с. 86-89.
5. Мирошник М. А. Отказоустойчивость распределенных телекоммуникационных систем. / Мирошник М.А. // Радиотехника: Всеукр. межвед. науч.-техн. сб. № 168. – 2012. – С. 51-55.

*Мирошник М. А., д.т.н., профессор (УкрГУЖТ),
Салфетникова Ю. Н., ассистент (НТУ «ХПИ»),
Пахомов Ю. В., ассистент (ХНУГХ)*

УДК 681.3

ПРОЕКТИРОВАНИЕ ЛОГИЧЕСКИХ БЛОКОВ УПРАВЛЕНИЯ С ПОМОЩЬЮ ШАБЛОНОВ ОПИСАНИЯ КОНЕЧНЫХ АВТОМАТОВ

В работе предложено использовать шаблоны автоматного программирования (patterns automata-based programming) для проектирования устройств логического управления на основе конечных автоматов. Для описания алгоритма функционирования автоматного устройства логического управления предложено использовать темпоральный граф переходов автомата (temporal state diagram), в котором учитываются задержки реального времени для каждого из состояний автомата. При проектировании конечного автомата на технологической платформе ПЛИС FPGA алгоритм функционирования описывается на языке описания аппаратуры VHDL, а синтез устройства осуществляется в САПР (CAD) XILINX ISE, а при проектировании конечного автомата на базе микроконтроллера семейства MCS 51 алгоритм функционирования описывается на подмножестве языка С в среде разработки Reil.

На основе общей концепции построения систем автоматизированного управления в них всегда можно выделить управляющие устройства и управляемые объекты. Следуя этой концепции, системы управления на основе конечных автоматов (finite state machines) также можно разделить на две части: управляющую часть, ответственную за логику поведения – выбор выполняемых действий, зависящий от текущего состояния и входного воздействия, а также за переход в новое состояние; управляемую часть, ответственную за выполнение действий, выбранных для выполнения управляющей частью, и, возможно, за формирование некоторых компонентов входных воздействий для управляющей части – обратных связей [1].

Среди всего множества управляющих устройств можно выделить устройства логического управления, у которых управляющие воздействия (control value) представляются в двоичном алфавите. Поскольку для реализации управляющей части в таких устройствах, как правило, используются конечные автоматы, то они называются управляющие автоматы. Подобные устройства широко применяются в системах Internet of [2].

Любое локальное цифровое устройство, реализующее алгоритм обработки информации или управления, может быть реализовано двумя способами: аппаратным или программно-аппаратным [3].

При аппаратном способе реализации заданный алгоритм описывается на языке описания аппаратуры (HDL) и синтезируется инструментальными средствами систем автоматизированного проектирования (САПР), а затем имплементируется в ПЛИС (программируемые логические интегральные схемы) или ASIC (application-specific integrated circuit или интегральная схема специального назначения). Достоинством такого подхода является аппаратная гибкость (возможность реализовать любой алгоритм) и достаточно большое быстродействие. К недостаткам данного подхода можно отнести необходимость разработки интерфейса ввода-вывода для связи разрабатываемого устройства с внешними устройствами и сложности в реализации временных параметров [4].

При программно-аппаратном способе реализации алгоритм описывается на аппаратно-ориентированном языке программирования (например, на языке С со специальными библиотеками) с учетом аппаратной архитектуры, на которой будет реализовываться заданная программа. Это, как правило, различные семейства микроконтроллеров (МК). Достоинством данного подхода является наличие специальных аппаратно-ориентированных функций (таймеров контроллеров прерываний), а также наличие аппаратно реализованных интерфейсов обмена с внешними устройствами. К недостаткам можно отнести меньшее