

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

быстродействие и ограничения, накладываемые аппаратной архитектурой на реализуемый алгоритм [5].

При описании алгоритма функционирования цифровых устройств логического управления в САПР цифровых устройств одним из стилей написания кода является стиль автоматного программирования. Суть автоматного программирования состоит в отделении описания логики поведения (при каких условиях необходимо выполнить те или иные действия) от описания его семантики (собственно смысла каждого из действий). В автоматном программировании в качестве базового используется понятие «состояние», а не «класс», «объект», и др., а в качестве описания алгоритма функционирования используется граф переходов [6].

Автоматные программы строго структурированы и в них выделены три вида функций: функции переходов, функции выходов, функции реализации задержек и перехода в новое состояние. Автоматные программы строго шаблонизированы с использованием операторов многопозиционного выбора (switch, case), условных операторов (if, select) и функций реализации таймера или фронта (синхросигнала Clk). Автоматные программы инвариантны к способу кодирования (языку программной реализации). Есть примеры автоматных программ на разных языках описания аппаратуры, на C, на JavaScript и др. [7].

Таким образом, актуальной становится задача разработки единого шаблона языкового описания автоматных устройств логического управления в стиле автоматного программирования. Целью данной работы является разработка шаблонов описания конечных управляющих автоматов на языках программирования и описания аппаратуры с последующей схемной реализацией разработанных программных кодов [8].

В результате проведенных исследований было показано, что использование стиля автоматного программирования и шаблонов описания конечных автоматов (finite state machines patterns) эффективно при написании программного кода как на языке описания аппаратуры VHDL, так и на языке программирования C. Схемная реализация разработанного кода инструментальными средствами соответствующих САПР подтвердило работоспособность предложенного метода.

Определенным положительным аспектом предложенного подхода является возможность верификации разработанного кода и диагностирования полученных схем на основе методик экспериментов над конечными автоматами путем разных вариантов обхода графа переходов.

Список литературы

1. Шалыто А.А. Автоматное программирование / Н.И. Поликарпова, А.А. Шалыто. – СПб.: Питер, 2011. – 167 с.
2. Baranov S. Logic and System Design of Digital Systems / S. Baranov. – Tallinn: TUT Press, 2008. – 267 p.
3. Lee, I. Handbook of real-time and embedded systems / Insup Lee, Joseph Y-T. Leung, Sang H. Son. – USA, Chapman & Hall/CRC, 2007. – 689 p.
4. Hahanov V. Cloud-driven traffic control: Formal modeling and technical realization /V. Hahanov; A. Zhalilo; W. Gharibi; E. Litvinova // 2015 4th Mediterranean Conference on Embedded Computing (MECO). 2015. – P: 21-24.
5. Shkil A.S. Design automation of easy-tested digital finite state machines / М.А. Miroshnyk, Y.V. Pakhomov, A.S. Shkil, E.N. Kulak, D.Y. Kucherenko // Radio Electronics, Computer Science, Control. – 2018. – №2. – P. 117-124.
6. Haskell R. Digital Design Using Digilent FPGA Boards - VHDL / Active-HDL Edition / Richard E. Haskell, Darrin M. Hanna. – LBE Books Rochester Hills, MI, 2009. – 381 p.
7. Shalyto A. State Machine Design Pattern / A. Shalyto, N. Shamgunov, G. Korneev // .NET Technologies 2006 – Shot communication papers conference proceedings. 4-th International Conference in Central Europe on .Net Technologies. University of West Bohemia. May 29 – June 1, 2006 – P. 51-57.
8. Матюшин А. О. Программирование микроконтроллеров: Стратегия и тактика / А.О. Матюшин. – М.: ДМК Пресс, 2017. – 356 с.

Доценко С. І., д.т.н., доцент (УкрДУЗТ)

МЕТОДОЛОГІЯ ЦІЛІСНОГО ПІДХОДУ ДО ДОСЛІДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ: АНТИНОМІЇ ЦІЛІСНОСТІ (ЧАСТИНА ДРУГА)

На основі виконаного в [1] аналізу проблем системного та цілісного підходів до дослідження інтелектуальних систем, встановлена наявність серії антиномій цілісності. При цьому, антиномії з першої по десятю були досліджені і розкриті у роботі [2].

Антиномії з одинадцятої по двадцять четверту потребують свого дослідження і послідуочого розкриття.

В той же час, подальше дослідження інтелектуальних систем з застосуванням методології цілісного підходу забезпечило встановлення додаткових антиномій, які стосуються управлінського аспекту їх діяльності.

Протиріччя 25 – антиномія управління підприємством: в теорії менеджменту первинним є