

УДК 621.372.8:621.317.7

МИРОШНИК М. А., д. т. н., профессор кафедры специализированных компьютерных систем (Украинский государственный университет железнодорожного транспорта),  
КЛИМЕНКО Л. А., к. т. н., доцент кафедры специализированных компьютерных систем (Украинский государственный университет железнодорожного транспорта),  
ПАХОМОВ Ю. В., ассистент (Харьковский национальный университет городского хозяйства имени А.Н. Бекетова)

## Методы автоматизации проектирования легкотестируемых компьютерных систем и устройств на основе цифровых автоматов

*Представлены усовершенствованные методы проектирования легкотестируемых цифровых устройств и математическая модель конечных управляющих автоматов на языках описания аппаратуры. Актуальность работы состоит в обеспечении минимальных дополнительных аппаратурных затрат при автоматизированном проектировании легкотестируемых цифровых устройств, представленных моделями конечных управляющих автоматов на языках описания аппаратуры. Цель работы состоит в разработке процедуры построения моделей легкотестируемых управляющих автоматов на языках описания аппаратуры и оценке аппаратурных затрат для разных способов введения аппаратурной избыточности в HDL-модели автоматов.*

**Ключевые слова:** управляющий автомат, таблица переходов-выходов, сдвиговый регистр, сканируемый путь, язык описания аппаратуры, САПР, Active-HDL, XILINX ISE.

### Постановка проблемы в общем виде и ее связь с важными научными или практическими заданиями

При высокоуровневом проектировании цифровых устройств управления на основе конечных автоматов формой представления спецификации проектируемого устройства является таблица переходов-выходов (state table) или граф переходов автомата (state diagram). Одним из способов описания моделей ЦУ в форме конечных автоматов на языке VHDL является автоматный шаблон, т. е. способ описания моделей управляющих конечных автоматов, спецификация на которые задана в виде ТПВ или графа переходов. Это специальная структура VHDL-модели, в которой функции переходов и выходов выделены в отдельные процессы (процесс), а назначение нового состояния осуществляется в специальном процессе, связанном с синхронизацией. При проектировании тестопригодных управляющих автоматов аппаратурную избыточность, обеспечивающую легкотестируемость, целесообразно вносить еще на начальном этапе проектирования, т. е. при построении HDL-моделей проектируемых устройств. Легкотестируемым будем называть конечный автомат, для которого можно построить диагностический эксперимент минимальной длины путем обеспечения установки автомата в любое состояние за минимальное число тактов.

Таким образом, дана функциональная модель абстрактного автомата в виде ТПВ или графа переходов и на его основе строится VHDL-модель в форме автоматного шаблона. Необходимо рассмотреть различные способы внесения аппаратурной избыточности в VHDL-модель для обеспечения легкотестируемости и выбрать оптимальный способ с точки зрения дополнительных аппаратурных затрат. Аппаратурная избыточность в VHDL-моделях обеспечивается путем внесения в HDL-код дополнительных условных операторов, обеспечивающих построение сканируемого пути в запоминающей части автомата, что подтверждается результатами автоматизированного синтеза. Оптимальным с точки зрения внесения дополнительных аппаратурных затрат будем считать тот способ, который обеспечивает минимальные дополнительные аппаратурные затраты при оценке по Квайну вентильного эквивалента схемы, синтезируемой в кристаллы ПЛИС в автоматическом режиме инструментальными средствами систем автоматизированного проектирования (САПР) ЦУ.

Научная новизна работы состоит в дальнейшем развитии метода повышения тестопригодности конечных автоматов за счет расширения входного алфавита в HDL-моделях в форме автоматного шаблона, что дало возможность автоматизировать процесс проектирования легкотестируемых автоматов с использованием языков описания аппаратуры.

© М. А. Мирошник, Л. А. Клименко, Ю. В. Пахомов, 2018

Практическая ценность полученных результатов заключается в разработке процедур внесения избыточности и расширения входного алфавита в HDL-моделях конечных автоматов в форме автоматного шаблона путем внесения дополнительных условных операторов в HDL-код, которые обеспечивают установку автомата в произвольное состояние. Разработанные процедуры могут быть применены при разработке дополнительного программного модуля САПР цифровых устройств, который будет в автоматизированном режиме формировать HDL-код легкотестируемого конечного автомата.

#### **Анализ последних исследований и публикаций**

Методы повышения тестопригодности ЦУ путем внесения в схемную реализацию аппаратурой избыточности достаточно развиты и повсеместно используются при проектировании. В классической работе по разработке и классификации структурных методов тестопригодного проектирования [1] введены понятия «управляемости» и «наблюдаемости» как основы оценки структурной тестопригодности. Изложены принципы организации сдвиговых регистров в запоминающей части ЦУ и построение на их основе сканируемого пути. Кроме того, предложены модификации известных структурных методов построения тестов при использовании методов сканирования. Практические рекомендации по использованию структурных методов повышения тестопригодности ЦУ и построению алгоритмов диагностирования на их основе подробно изложены в [2, 3].

В классической работе [4] рассмотрены методы встроеного самотестирования как основа обеспечения тестопригодности проектируемых устройств. Показано применение технологии BIST (built-in self-test) при проектировании цифровых систем на программируемых логических интегральных схемах (ПЛИС). В работе [5] получили дальнейшее развитие идеи тестопригодного проектирования на основе технологии граничного сканирования (Boundary Scan) с использованием порта JTAG. Рассмотрены стандарты граничного сканирования IEEE 1149.x, IEEE 1500 и IEEE 1532, а также стандарт тестопригодного проектирования IEEE 1687. Технологии тестопригодного проектирования и тестирования цифроаналоговых схем предложены в [6]. Рассмотрены методы функциональной декомпозиции схем, обеспечивающие автоматизированное построение тестопригодных цифроаналоговых структур.

Функциональные методы тестопригодного проектирования ЦУ подробно рассмотрены в [7]. Для цифровых автоматов, представленных в форме ТПВ, введены понятия диагностируемого и определенно-

диагностируемого классов автоматов и предложены способы приведения ТПВ автоматов к указанным классам. Рассмотрены процедуры проведения ДЭ с автоматами с использованием установочных, синхронизирующих, диагностических и переводящих последовательностей. Кроме того, обоснована идея повышения тестопригодности автомата за счет внесения аппаратурной избыточности путем расширения входного алфавита, выходного алфавита и алфавита состояний. Вопросы проведения ДЭ с автоматами путем обхода всех вершин и дуг графа переходов рассмотрены в [8]. Даны оценки длины и полноты ДЭ при различных вариантах обхода графа. Предложены процедуры повышения тестопригодности автоматов путем эквивалентных преобразований и расширения входного алфавита.

В [9] рассмотрены структурные модели конечных автоматов, которые позволяют использовать значения выходных переменных в качестве кодов внутренних состояний. Данный подход применен при синтезе конечных автоматов Мили на ПЛИС, что позволило снизить стоимость реализации в 1,5-2 раза. Также предложенный способ кодирования состояний позволяет снизить потребляемую мощность схемных реализаций конечных автоматов [10]. В [11] изложены методы тестопригодного проектирования ЦУ на основе ПЛИС и проведение ДЭ в сетях клеточных автоматов. Вопросы автоматизации проведения ДЭ над HDL-моделями конечных автоматов в форме автоматного шаблона рассмотрены в [12]. В [13] рассмотрены процедуры организации ДЭ в системе верификации САПР ПЛИС и предложены процедуры локализации ошибок проектирования в HDL-моделях.

#### **Выделение не решенных ранее частей общей проблемы, которым посвящается данная статья**

Актуальной является задача разработки процедур автоматизации проектирования легкотестируемых конечных автоматов, представленных на языках описания аппаратуры, сравнения дополнительных аппаратурных затрат при структурных и функциональных методах повышения тестопригодности на основе сравнения результатов синтеза тестопригодных HDL-моделей средствами САПР ПЛИС.

#### **Формулирование целей статьи (постановка задачи)**

Цель данной работы – разработать процедуры построения моделей легкотестируемых управляющих автоматов на языках описания аппаратуры и оценить аппаратурные затраты для разных способов введения аппаратурной избыточности в HDL-модели автоматов.

**Изложение основного материала исследования с полным обоснованием полученных научных результатов**

В качестве абстрактной модели ЦУ с памятью будем использовать конечный автомат, определяемый пятеркой  $W = \langle X, A, Y, \delta, \lambda \rangle$ , где  $X = \{x_1, x_2, \dots, x_m\}$  – множество букв входного алфавита;  $A = \{a_1, a_2, \dots, a_n\}$  – множество состояний автомата;  $Y = \{y_1, y_2, \dots, y_r\}$  – множество букв выходного алфавита;  $\delta(a_i, x_k) = a_j$  – функция переходов автомата,  $\lambda(a_i, x_k) = y_\alpha$  – функция выходов автомата.

Схема является тестопригодной, если процедура генерации множества тестовых наборов, оценка их эффективности и реализация тестового диагностирования могут быть выполнены при условии соблюдения в установленных пределах финансовых затрат, затрат времени и значений показателей, характеризующих приспособленность схемы к обнаружению неисправностей, поиску места неисправностей и реализации тестового диагностирования [1].

Экспериментом с автоматом будем называть процесс приложения входных последовательностей, наблюдения соответствующих им выходных последовательностей и вывода заключений, основанных на этих наблюдениях. В зависимости от цели эксперимента различаются эксперименты по идентификации состояний автомата, идентификации входной последовательности автомата и идентификации автомата с  $n$  состояниями, отличающегося от всех других автоматов с таким же числом состояний. С целью обнаружения ошибок проектирования в моделях управляющих автоматов, представленных ТПВ или графом переходов автомата, применяется диагностический эксперимент, связанный с обходом всех вершин графа. С этой точки зрения легкотестируемым является автомат, который может быть установлен в любое состояние не более, чем за  $n$  тактов, где  $n$  – число состояний автомата без использования синхронизирующих последовательностей [14].

Если рассматривать в качестве объекта диагностирования HDL-модели конечных автоматов в форме автоматного шаблона, то для построения на их основе легкотестируемых автоматов рассматриваемые модели должны отвечать следующим условиям:

- в модели должен быть специальный механизм перевода автомата в режим тестирования и, наоборот, на любом такте работы автомата;
- в режиме тестирования автомат может быть установлен в любое состояние за  $(n-1)$  тактов, где  $n$  –

число состояний автомата, и может быть организован гамильтонов цикл для любого из состояний автомата;

- легкотестируемый автомат должен строиться в автоматизированном режиме средствами САПР.

Основной принцип методов структурного тестопригодного проектирования схем состоит в организации сканирования запоминающей части ЦУ или просто «сканирования» [15]. Метод сканируемого пути позволяет решать проблему тестопригодности путем уменьшения сложности структуры схемы. Принцип, лежащий в основе метода, состоит в следующем:

- 1) элементы памяти проверяются отдельно от всей схемы и в первую очередь;

- 2) комбинационная часть (КЧ) проверяется во вторую очередь с возможностью установки внутренних переменных в любое состояние (независимо от их предыдущего состояния) и наблюдения выходов КЧ непосредственно.

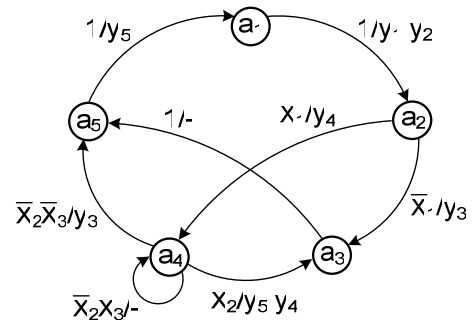
При реализации метода сканируемого пути каждому элементу памяти предшествует мультиплексор «2-1», управляемый общим сигналом выбора режима сканирования. Таким образом, в режиме сканирования элементы памяти можно достаточно просто установить в любое заданное множество состояний, подавая последовательность сигналов на вход сканируемых данных и тактируя сдвиговый регистр с помощью синхроимпульса.

В качестве примера преобразования HDL-моделей управляющих автоматов рассмотрим модель автомата Мили на языке VHDL. На рис. 1 представлены таблица переходов-выходов (а) и граф переходов (б) данного автомата.

В дальнейшем (если не оговорено особо) будем рассматривать автоматы, работа которых тактируется сигналами от генератора синхронизирующих импульсов (CLK), т.е. в автомате кроме входов  $(x_1, \dots, x_n)$  имеется по крайней мере еще один вход CLK, по которому поступает синхронизирующая последовательность. Поэтому входным сигналом на переходе  $(a_m, a_s)$ , соответствующим пути  $a_m X(a_m, a_s) - Y(a_m, a_s) a_s$ , будет не  $X(a_m, a_s)$ , а конъюнкция  $CLK \cdot X(a_m, a_s)$ . Таким образом, при  $X(a_m, a_s) = 1$  (безусловный переход) автомат работает только от синхроимпульса. Но при построении графа переходов автомата символ CLK дугам графа переходов, как правило, не приписывается.

a	1	$x_1$	$\overline{x_1}$	$x_2$	$\overline{x_2}x_3$	$\overline{x_2}\overline{x_3}$
a <sub>1</sub>	a <sub>2</sub> /y <sub>1</sub> ,y <sub>2</sub>					
a <sub>2</sub>		a <sub>3</sub> /y <sub>3</sub>	a <sub>4</sub> /y <sub>4</sub>			
a <sub>3</sub>	a <sub>5</sub> /-					
a <sub>4</sub>				a <sub>3</sub> /y <sub>4</sub> ,y <sub>5</sub>	a <sub>4</sub> /-	a <sub>5</sub> /y <sub>3</sub>
a <sub>5</sub>	a <sub>1</sub> /y <sub>5</sub>					

a)



б)

Рис. 1. Таблица переходов-выходов и граф переходов автомата Мили

Рассмотрим структурный способ организации сканируемого пути в HDL-моделях конечных автоматов путем непосредственного введения сдвигового регистра в HDL-модель вручную. Но данная модель несколько отличается от автоматного шаблона, т. к. коды состояний автомата при этом должны задаваться принудительно.

На рис. 2 приведен фрагмент VHDL-модели автомата Мили в форме автоматного шаблона [12] со сдвиговым регистром в запоминающей части. Сигнал A=1 задает режим стандартного функционирования автомата, A=0 – режим сдвига в регистре с внешнего

входа TDI (test data input) в сторону старших разрядов. Состояния автомата, в данной модели, кодируются принудительно по номеру состояния (при декларации внутренних сигналов состояний автомата) с использованием минимального числа триггеров (в данном случае 3) для хранения кода состояния. Такая организация модели дает возможность записывать извне с входа TDI (test data input) код нужного состояния за минимальное число тактов, определяемое числом разрядов сдвигового регистра.

```

architecture FSM_MX of FSM_Mealy_MX is
    signal State, NextState: STD_LOGIC_vector (2 downto 0);
    signal a1: STD_LOGIC_vector (2 downto 0):="001";
        signal a2: STD_LOGIC_vector (2 downto 0):="010";
        signal a3: STD_LOGIC_vector (2 downto 0):="011";
        signal a4: STD_LOGIC_vector (2 downto 0):="100";
        signal a5: STD_LOGIC_vector (2 downto 0):="101";
begin
    Sreg0_CurrentState: process (Clk, Reset)
    begin
        if Reset='1' then State <= a1;
        elsif Clk'event and Clk = '1' then
            if A='1' then State <= NextState;
            else State <= State(1 downto 0) & TDI;
            end if;
        end if;
    end process;

```

Рис. 2. Фрагмент архитектуры VHDL-модели автомата Мили со сдвиговым регистром (FSM\_Mealy\_MX)

На рис. 3 приведена временная диаграмма (waveform) моделирования работы автомата Мили со сдвиговым регистром в запоминающей части и внешним входом TDI (FSM\_Mealy\_MX). Сигнал State приведен на диаграмме в десятичном виде. При A=1

(от 0 нс до 500 нс) автомат работает в нормальном режиме, при A=0 (от 500 нс до 2000 нс) – в сдвиговом режиме, при этом значение TDI записывается в старшие разряды.

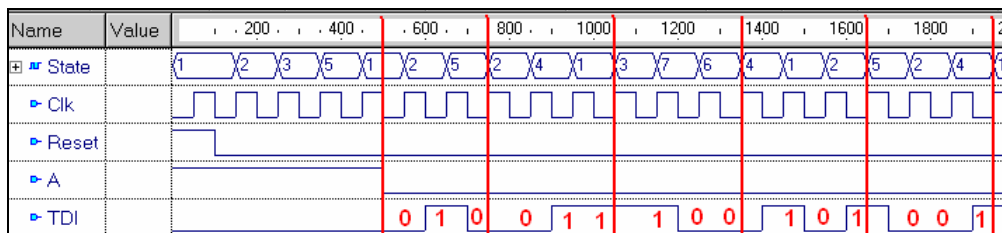


Рис. 3. Временная диаграмма работы автомата Мили со сдвиговым регистром (FSM\_Mealy\_MX)

На рис. 4 приведен фрагмент схемы автомата Мили со сдвиговым регистром и внешним входом TDI (FSM\_MX) после синтеза в системе Xilinx ISE. Полное уравнение для триггера D2 будет:

$$D_2 = \{[(x_3 \vee \bar{x}_3) y_3 x_2 \vee (x_1 y_1 \vee y_2)] A \vee Q_2 \bar{A}\} \overline{Reset}$$

Данный фрагмент синтезируемой схемы фактически является мультиплексором относительно управляющего сигнала A, что полностью соответствует структуре схемы со сканированием пути.

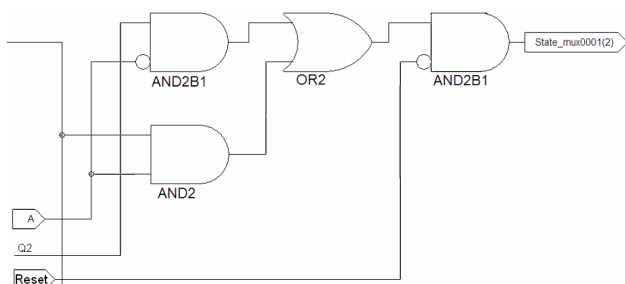


Рис. 4. Фрагмент схемы автомата Мили со сдвиговым регистром (FSM\_Mealy\_MX)

Для организации гамильтонова цикла обхода графа может быть применен регистр циклического сдвига. Предлагается следующая реализация автомата Мили с циклическим сдвиговым регистром в запоминающей части, приведенная на рис. 5: сигнал A=1 задает режим «нормальной» работы автомата (режим параллельной записи в регистр), A=0 – режим циклического сдвига в регистре от младших разрядов к старшим (влево). Состояния автомата в данном модуле кодируются принудительно унитарно, т.е. на каждое состояние выделяется отдельный триггер: a1 (00001), a2 (00010), a3 (00100), a4 (01000), a5 (10000).

Такая организация модели дает возможность с каждым новым тактом переходить в новое состояние и в сдвиговом режиме обходить все состояния за число тактов, равное числу состояний автомата.

```

if A='1' then State <= NextState;
else State <= State(3 downto 0) & State(4);
end if;
    
```

Рис. 5. Реализация автомата Мили с циклическим сдвиговым регистром в запоминающей части

Данный подход достаточно эффективно решает задачу построения гамильтонова цикла обхода графа в режиме диагностирования, но за счет унитарного кодирования состояний приводит к значительным аппаратным затратам и снижению быстродействия.

С одной стороны, предложенная методология структурного преобразования автоматов в легкотестируемые за счет введения сдвиговых регистров решает поставленную задачу, но с другой стороны – имеет ряд недостатков.

1. Появление дополнительного входа TDI является полезным при организации структуры Boundary Scan, но создает дополнительные трудности при организации работы автомата, т.к. в стандартном режиме работы автомата управление данным входом не предусмотрено.

2. Разомкнутый сдвиговый регистр создает дополнительные сложности для организации гамильтонова цикла при сканировании автомата.

3. Использование сдвигового регистра при произвольном кодировании состояний автомата снижает быстродействие в режиме сканирования.

4. Применение циклического сдвигового регистра с унитарным кодированием состояний решает задачу гамильтонова цикла, но существенно повышает аппаратные затраты.

Таким образом, предложенная методика не решает всех поставленных задач, требуется иное решение. Оно находится в области функционального преобразования ТПВ автомата путем расширения входного алфавита для обеспечения тестопригодности.

При функциональном подходе к проектированию (анализу) цифровых устройств, заданных моделью конечного автомата  $W = \langle X, A, Y, \delta, \lambda \rangle$ , повышение тестопригодности автомата возможно только за счет расширения входного алфавита X, алфавита состояний A или выходного алфавита Y [7]. С одной стороны, это приводит к внесению в схему проектируемого устройства аппаратной избыточности, а с другой – обеспечивает сохранение алгоритма функционирования автомата, заданного, как правило, ТПВ автомата или его графом переходов.

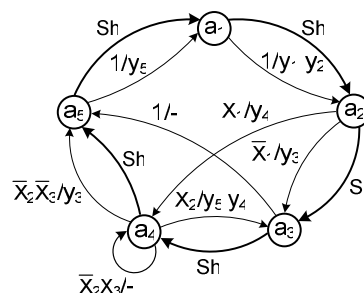
Авторами данной работы в [14] предложено и теоретически обосновано введение дополнительного столбца (символа) в таблицу переходов-выходов

автомата, забезпечуючого для цього символа функцію переходов графа переходов сдвигового регистра, що дозволяє установити автомат в будь-яке заданне состояние. На основани указаних теоретических положений расширим ТПВ автомата Мили (рис. 1) добавлением столбца Sh (Shift). При

Sh=1 автомат работает в режиме установки в любое заданное состояние, а при Sh=0 автомат реализует заданный алгоритм. Расширенная ТПВ и модифицированный граф переходов автомата Мили показаны на рис. 6.

a	1	$x_1$	$\overline{x_1}$	$x_2$	$\overline{x_2 x_3}$	$\overline{x_2 x_3}$	Sh
a <sub>1</sub>	a <sub>2</sub> / y <sub>1</sub> ,y <sub>2</sub>						a <sub>2</sub> /-
a <sub>2</sub>		a <sub>3</sub> / y <sub>3</sub>	a <sub>4</sub> / y <sub>4</sub>				a <sub>3</sub> /-
a <sub>3</sub>	a <sub>5</sub> /-						a <sub>4</sub> /-
a <sub>4</sub>				a <sub>3</sub> / y <sub>4</sub> ,y <sub>5</sub>	a <sub>4</sub> /-	a <sub>5</sub> /y <sub>3</sub>	a <sub>5</sub> /-
a <sub>5</sub>	a <sub>1</sub> / y <sub>5</sub>						a <sub>1</sub> /-

а)



б)

Рис. 6. Расширенная ТПВ и модифицированный граф переходов автомата Мили с сигналом Sh

**Выводы из данного исследования и перспективы дальнейшего развития в данном направлении**

В работе представлен метод автоматизированного проектирования легкотестируемых управляющих автоматов путем внесения аппаратной избыточности. Модель автомата представлена на языке VHDL в форме автоматного шаблона. Способом решения является внесение дополнительных фрагментов VHDL-кода, обеспечивающих принудительную установку автомата в произвольное состояние без использования синхронизирующих последовательностей. Рассмотрено использование сдвигового регистра в запоминающей части управляющего автомата для организации сканирования пути. Предложен метод расширения таблицы переходов-выходов автомата, который обеспечивает режим обхода всех вершин графа переходов автомата (состояний) в режиме диагностирования. Указанный подход повышает управляемость состояний автомата, что значительно улучшает его тестопригодность. Моделирование расширенных VHDL-моделей автомата средствами Active-HDL подтвердило работоспособность данного подхода. Синтез данных моделей средствами САПР XILINX ISE подтвердил получение тестопригодных структур и показал минимальные аппаратные затраты для метода, связанного с расширением таблицы переходов-выходов, по сравнению с организацией сдвигового регистра в режиме Scan Path.

**Результаты**

Моделирование расширенных HDL-моделей управляющего автомата средствами Active-HDL подтвердило работоспособность данного подхода. Синтез данных моделей инструментальными средствами автоматизированного проектирования

XILINX ISE подтвердил получение тестопригодных структур и показал минимальные аппаратные затраты для метода, связанного с расширением таблицы переходов-выходов, по сравнению с организацией сдвигового регистра в режиме Scan Path.

**Выводы**

В работе решена задача автоматизированного проектирования тестопригодных управляющих автоматов на основе применения методов установки автоматов в заданное состояние. Оптимальным с точки зрения аппаратных затрат способом организации установки в произвольное состояние управляющих автоматов является расширение таблицы переходов-выходов, которое повышает управляемость состояний автомата и приводит к преобразованию структуры их HDL-моделей в легкотестируемые.

Научная новизна работы состоит в дальнейшем развитии метода повышения тестопригодности конечных автоматов за счет расширения входного алфавита в HDL-моделях в форме автоматного шаблона, что дало возможность автоматизировать процесс проектирования легкотестируемых автоматов с использованием языков описания аппаратуры.

Практическая ценность полученных результатов заключается в разработке процедур внесения избыточности и расширения входного алфавита в HDL-моделях конечных автоматов в форме автоматного шаблона путем внесения дополнительных условных операторов в HDL-код, которые обеспечивают установку автомата в произвольное состояние. Разработанные процедуры могут быть применены при разработке дополнительного программного модуля САПР цифровых устройств, который будет в автоматизированном режиме формировать HDL-код легкотестируемого конечного автомата.

## Список использованных источников

1. Беннеттс, Р. Дж. Проектирование тестопригодных логических схем [Текст]: пер. с англ. / Р. Дж. Беннеттс. – М.: Радио и связь, 1990. – 176 с.
2. Пархоменко, П. П. Основы технической диагностики (Оптимизация алгоритмов диагностирования, аппаратурные средства) [Текст] / П. П. Пархоменко, Е. С. Согомоян; под ред. П. П. Пархоменко. – М.: Энергия, 1981. – 320 с.
3. Горяшко, А. П. Проектирование легкотестируемых дискретных устройств: идеи, методы, реализация [Текст] / А. П. Горяшко // Автоматика и телемеханика. – 1984. – № 7. – С. 5–35.
4. Stroud, C.E. A designer's guide to built-in self-test [Text] / Charles E. Stroud. – Kluwer Academic Publishing, 2002. – 319 p.
5. Городецкий, А. Введение в технологии JTAG и DFT. Тестирование в технологиях граничного сканирования и тестопригодное проектирование [Текст] / Ами Городецкий. – Palmarium Academic Publishing, Germany, 2012. – 308 с.
6. Mosin, S. Methodology to Design-For-Test Automation for Mixed Signal Integrated Circuit [Text] / S. Mosin // Proceedings of the International Symposium. EWDTS'2013, September 27-30, Rostov on Don, Russia, 2013. – P.178–183.
7. Тоценко, В. Г. Алгоритмы технического диагностирования цифровых устройств [Текст] / В. Г. Тоценко. – М.: Радио и связь, 1985. – 240 с.
8. Practical Methods for de Bruijn sequences Generation using Non-Linear Feedback Shift Registers [Text] / O. Demihev, M. Miroschnyk, D. Karaman, I. Filippenko, V. Krylova, T. Korytchinko // 14th IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, Lviv-Slavske, Ukraine, 2018/2. – P. 35.
9. Solov'ev, V. V. Minimization of mealy finite-state machines by using the values of the output variables for state assignment [Text] / V. V. Solov'ev // Journal of Computer and Systems Sciences International. – January 2017. – Volume 56, Issue 1. – P. 96–104.
10. Solov'ev, V. V. Minimization of Power Consumption of Finite State Machines by Splitting Their Internal States [Text] / V. V. Solov'ev, T. N. Grzes // Journal of Computer and Systems Sciences International. – 2015. – Vol. 54, No. 3. – P. 367–374.
11. Мирошник, М. А. Проектирование диагностической инфраструктуры вычислительных систем и устройств на ПЛИС [Текст]: монография / М. А. Мирошник. – Харьков: ХУПС, 2012. – 188 с.
12. Шкиль, А. С. Автоматизация поиска ошибок проектирования в HDL-моделях конечных автоматов [Текст] / А. С. Шкиль, Г. П. Фастовец, А. С. Серокурова // АСУ и приборы автоматики. – 2014. – Вып. 168. – С. 43-52.
13. Miroschnyk, M.A. Design automation of testable finite state machines [Text] / M. A. Miroschnyk, D. E. Kucherenko, Yu. V. Pahomov // 15th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS-2017). Харьковский национальный университет радиоэлектроники. – 2017. – P. 203-208.
14. Synchronizing Sequences For Verification Of Finite State Machines [Text] / A. S. Grebenyuk, M. A. Miroschnik, A. S. Shkil, E. N. Kulak, I. V. Filippenko, D. Y. Kucherenko // 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT'2018) May 24 - 27, 2018/5.
15. Методы синтеза легкотестируемых цифровых автоматов [Текст] / М. А. Мирошник, Ю. В. Пахомов А. С. Гребенюк, И. В. Филиппенко // Інформаційно-керуючі системи на залізничному транспорті. – 2016. – № 5. – С. 28-39.

**Мірошник М. А., Клименко Л. А., Пахомов Ю. В. Методи автоматизації проектування легкотестованих комп'ютерних систем і пристроїв на основі цифрових автоматів.** Подано вдосконалені методи проектування легкотестованих цифрових пристроїв і математична модель кінцевих керуючих автоматів мовами опису апаратури. Актуальність роботи полягає в забезпеченні мінімальних додаткових витрат апаратури при автоматизованому проектуванні легкотестованих цифрових пристроїв, представлених моделями кінцевих керуючих автоматів мовами опису апаратури. Мета роботи полягає в розробленні процедури побудови моделей легкотестованих керуючих автоматів мовами опису апаратури та оцінюванні апаратурних витрат для різних способів введення апаратурної надмірності в HDL-моделі автоматів.

**Ключові слова:** керуючий автомат, таблиця переходів-виходів, зсувний регістр, сканований шлях, мова опису апаратури, САПР, Active-HDL, XILINX ISE.

**Miroschnyk M. A., Klimenko L. A., Pahomov Yu. V. Methods for automating the design of easily tested computer systems and devices based on digital automata.** The relevance of the work is to provide minimal additional hardware costs during design automation of easy-tested digital devices, which are represented by models of control finite state machines on hardware description languages.

To develop procedures of models' constructing of easy-tested control finite state machines on hardware

description languages and estimate hardware costs for different methods of hardware redundancy introduction to HDL-models of finite state machines.

The introduction to HDL-models of control finite state machines, which are presented in the form of the FSM template, hardware redundancy (additional fragments of the HDL-code), providing the forcing setting of finite state machine into an arbitrary state without the use of synchronizing sequences. For implementation of this approach, the method of FSM's state table extending is applied, which ensures the mode of bypassing of all nodes of FSM' state diagram in the diagnostic mode.

Simulation of extended VHDL-models of the control FSM using Active-HDL confirmed the operability of this approach. Synthesis of these models using CAD XILINX ISE confirmed the receipt of testable structures and showed the minimum hardware costs for the method associated with the extension of the state table, in comparison with the organization of the shift register in the Scan Path mode.

The task of computer-aided design of testable control finite state machine on the basis of application of FSM' setting methods into given state is solved in the work. The optimal way of the setting organization into an arbitrary state of the control FSM is to expand the state table, which improves the controllability of FSM' states and leads to the structure' transformation of their HDL-models into easy-tested ones. The scientific novelty of the work is the transformation of control FSM' models on hardware description languages, which is realized by introduction of the additional symbol to the state table, providing the settings of the FSM into an arbitrary state without the use of synchronizing sequences. The practical significance of obtained results is to confirm the optimality, in terms of additional hardware costs, of the setting method of the control FSM into an arbitrary state by introducing the additional symbol into the state table.

**Keywords:** control finite state machine, state table, shift register, scanned path, hardware description language, CAD, Active-HDL, XILINX ISE.

*Пахомов Юрій Васильович* асистент, Харківський національний університет міського господарства імені О. М. Бекетова, Харків, Україна. E-mail: [abc050073@gmail.com](mailto:abc050073@gmail.com) <https://orcid.org/0000-0002-2267-8600>

*Miroshnyk Maryna Anatolijvna.*, doctor of technical science, professor of the Department of Computer Systems Ukrainian State University of Railway Transport, Kharkiv, Ukraine

E-mail: [miroshnyk@kart.edu.ua](mailto:miroshnyk@kart.edu.ua) <http://orcid.org/0000-0002-2231-2529>

*Liybov Klymenko*, Doctor of Philosophy, associate professor department of specialized computer systems, Ukrainian State University of Railway Transport, Kharkiv, Ukraine.

E-mail: [lyubasha@email.ua](mailto:lyubasha@email.ua). <http://orcid.org/0000-0002-8252-7425>

*Pakhomov Y.V.* assistant, Kharkiv National University of municipal economy, Kharkiv, Ukraine.

E-mail: [abc050073@gmail.com](mailto:abc050073@gmail.com). <https://orcid.org/0000-0002-2267-8600>

Надійшла 08.06.2018 р.

*Мірошник Марина Анатоліївна*, доктор технічних наук, професор кафедри спеціалізованих комп'ютерних систем, Український державний університет залізничного транспорту, Харків, Україна. E-mail: [miroshnyk@kart.edu.ua](mailto:miroshnyk@kart.edu.ua)  
ORCID ID [0000-0002-2231-2529](http://orcid.org/0000-0002-2231-2529)

*Клименко Любов Анатоліївна*, кандидат технічних наук, доцент кафедри СКС, Український державний університет залізничного транспорту, Харків, Україна. E-mail: [lyubasha@email.ua](mailto:lyubasha@email.ua)  
ORCID ID <http://orcid.org/0000-0002-8252-7425>