

*Мирошник М. А., д-р техн. наук, проф. (УкрГУЖТ),  
Шкіль А. С., к.т.н., доцент,  
Кулак Э. Н., к.т.н., доцент,  
Филиппенко И. В., к.т.н., доцент,  
Рахлис Д. Е., к.т.н., доцент,  
Пахомов Ю.В., к.т.н.  
(ХНУРЭ)*

## **АППАРАТНАЯ РЕАЛИЗАЦИЯ ВРЕМЕННЫХ КОНЕЧНЫХ АВТОМАТОВ**

Среди всего множества систем управления значительную часть составляют системы логического управления (Logical Control System), у которых управляющие сигналы принимают значения логического нуля или единицы в зависимости от граничных значений физических величин, определяющих данные параметры. Для технической реализации указанных систем наиболее подходящей является модель структурного конечного автомата (Finite State Machines, FSM), а визуальным представлением алгоритма функционирования является граф переходов (State Diagram). Отличительной особенностью конечных автоматов в системах логического управления является наличие среди входных сигналов (Input Values) не только сигналов объекта управления, но и внешних по отношению к управляемой системе сигналов (событий) внешнего мира (External Events), которые обеспечивают взаимодействие системы логического управления с внешней средой [1].

Большинство реальных систем логического управления являются системами реального времени и для их реализации принято использовать модель временного автомата (timed FSM) а для визуального представления темпоральный граф переходов (Temporal State Diagram). При автоматизированном синтезе временных автоматов с использованием языков описания аппаратуры для корректного синтеза схемы автомата с учетом временных параметров, как правило, используются шаблоны автоматного программирования на языках описания аппаратуры (HDL). Для языка VHDL это специальная структура VHDL-модели, в которой функции переходов и выходов выделены в отдельные процессы (процесс), назначение нового состояния осуществляется в специальном процессе, связанном с синхронизацией, а задержки реализуются через счетчик автоматных тактов. Кроме того, при обработке внешних событий в системах реального времени необходимо учитывать временной промежуток, в течении которого внешние событие может изменить алгоритм работы управляющего автомата.

Целью данной работы является разработка единого шаблона на языке VHDL для описания разных типов временных управляющих автоматов в системах

логического управления реального времени в стиле автоматного программирования, которые реализуются на аппаратной платформе FPGA.

Модели конечных управляющих автоматов, реализованные как в hardware, так и в software, широко используются в системах логического управления реального времени.

В [1] определено роль и место управляющих автоматов в системах логического управления. Приведена методика проектирования автоматных систем логического управления с учетом реального времени и обработки внешних событий. Приведена классификация внешних событий и способы их обработки.

Модель временного автомата (timed automata), как способ реализации алгоритма управления в системах реального времени введен в работе [2]. Графовая модель автомата дополняется конечным множеством таймеров, принимающих действительные значения. Вершины графа называются позициями, а ребра – переходами. Каждый таймер сбрасывается в ноль в момент перехода и увеличивает свое значение с каждым автоматным тактом. С каждым переходом связано временное ограничение (clock constraint), которое означает, что данный переход может быть осуществлен только в том случае, если текущие значения таймера удовлетворяют этому ограничению. С каждой позицией связано ограничение на таймеры, называемое инвариантом; система может находиться в данной позиции только до тех пор, пока выполняется ее инвариант.

Дальнейшее развитие теория временных автоматов получила в работах, связанных с тестированием аппаратных систем реального времени. В [3] рассматриваются модели временных FSM (TFSM), учитывающие timeouts в состояниях и задержки выходных сигналов по отношению к реализации перехода в состояние. При этом учитывается, что если в течение timeout не поступило никакого входного сигнала, то автомат переходит в следующее состояние. Рассмотрены методы построения тестов для композиции TFSM. Рассматриваемые модели применялись при тестировании временных параметров клиент-серверных банковских систем. В [4] рассмотрены вопросы построения тестов для композиции временных автоматов с временными ограничениями (timed guards) и выходными задержками (outputs delay). В этой статье обсуждается, как можно сократить набор тестов с гарантированным покрытием отказов для системы взаимодействующих TFSM, при некоторых неисправных компонентах. При введении соответствующего теста, учитывая компоненты TFSM предполагаем, что все остальные компоненты исправны. В [5] предложены методы минимизации моделей временных автоматов для систем с таймаутами и временными ограничениями.

Также рассмотрены частные случаи использования моделей TFSM только с таймаутами или только с временными ограничениями. Рассмотрены вопросы минимизации не только числа состояний, но и временных характеристик. В [6] рассмотрена обобщенная модель временного автомата с таймаутами и временными ограничениями/ и выходными задержками. Также была получена процедура построения минимальной формы для детерминированных TFSM, которая уменьшает количество состояний, переходов и значений тайм-аута в каждом состоянии, и она уникальна с точностью до изоморфизма для неинициализированных TFSMs.

В работе [7] рассмотрено влияние способов кодировки состояний в структурных моделях конечных автоматов на аппаратные затраты и быстродействие. Предложен подход, который позволяет триггеры входных и выходных буферов ПЛИС применять в качестве элементов памяти конечного автомата. С этой целью предложена новая классификация структурных моделей конечных автоматов, согласно которой все конечные автоматы делятся на шесть классов: A, B, C, D, E и F. В моделях автоматов классов C и D в качестве элементов памяти используются триггеры выходных буферов, а в моделях автоматов классов E и F входных буферов. Это позволило уменьшить затраты внутренних элементов памяти в среднем на 25%.

В [9] введена новая система классификации моделей конечных автоматов по реализации переходов и выходных сигналов. Все автоматы делятся на три категории. У автоматов первой категории (регулярных, regular) переходы зависят только от входных сигналов, а значения выходных сигналов зависят только от состояний. У автоматов второй категории (временных, timed) переходы зависят от входных сигналов и времени их появления, а значения выходных сигналов зависят только от состояний. У автоматов третьей категории (рекурсивных, recursive) переходы зависят от входных сигналов и времени их появления, а значения выходных сигналов зависят от текущего состояния и предыдущего состояния непосредственно, т.е. для выходных сигналов состояния  $a_i$  реализуется функция  $y_i = y_i + u_j$ , где  $a_j$  предыдущее состояние автомата. Для данных категорий автоматов приведены различные шаблоны HDL-моделей на языках VHDL и Verilog, а также результаты их моделирования в системе ModelSim (from Mentor Graphics) и синтеза с использованием Xilinx ISE.

При автоматизированном проектировании устройств управления в системах логического управления на технологической платформе ПЛИС используются модели на языках описания аппаратуры. В работе [10] предложен подход к построению VHDL-моделей временных автоматов Мура с учитывающих влияние внешних событий.

Для аппаратной реализации устройств логического управления используется модель структурного автомата. В структурных моделях входной алфавит  $X$  абстрактного автомата преобразуется в множество входных воздействий (set input value), алфавит состояний  $A$  преобразуется во множество внутренних переменных  $Z$ , а выходной алфавит  $Y$  – во множество выходных значений (set input value). При этом все три множества являются конечными. В системах реального времени структурный автомат представляется моделью временного автомата (timed FSM).

Состояние структурного управляющего автомата в системах логического управления характеризуется совокупностью выходных управляющих сигналов, которые появляются в определенные моменты времени и имеют определенную длительность. Переходы из состояния в состояние определяются типом входных сигналов и временем их появления. По способу формирования выходных сигналов управляющие автоматы классифицируются на модели Мура и Мили, а по способу обработки входных сигналов на активные и пассивные.

У временного автомата Мура переход из состояния в состояние осуществляется «мгновенно», а выходные сигналы в состоянии формируются с учетом задержек их появления. При нулевой задержке выходные сигналы появляются в момент входа автомата в состояние и не изменяются до конца нахождения автомата в данном состоянии.

У временного автомата Мили время появления выходных сигналов определяется временем появления входных сигналов (с учетом задержек). Время существования выходных сигналов автомата Мили определяется длительностью перехода в новое состояние и в момент входа в новое состояние выходные сигналы данного состояния обнуляются.

Если у автомата одновременно присутствуют выходные сигналы, характерные для моделей Мура и Мили, то такой автомат называется совмещенным (C-автомат) [10].

Активный временной автомат функционирует в зависимости от значения входного сигнала в определенный момент времени (не от изменения входного сигнала, т.е. входного события). Изменение входного сигнала (входное воздействие) непосредственно не инициирует изменения состояния автомата. Автомат опрашивает входные сигналы в моменты времени, определяемые алгоритмом его работы и, таким образом, реализует функцию переходов. Время перехода в новое состояние у активного автомата фиксированное (определяемое различными задержками в различных переходах), т.е. статическое.

Работа пассивного автомата определяется входным событием (автомат реагирует не на значения входного сигнала в определенный момент времени, а на событие

в определенный промежуток времени). Изменение входного сигнала (событие) непосредственно инициирует переход автомата в новое состояние (реализует функцию переходов) и определяет момент появления выходного сигнала (реализует функцию выходов). С этой точки зрения пассивный автомат можно определить как событийный. Время перехода в новое состояние у пассивного автомата не фиксированное, т.е. динамическое. Оно зависит от времени появления входного события, которое по сути является случайным событием, во временном промежутке ожидания данного события.

Микропрограммный автомат – всегда активный автомат: с каждым новым тактом Clk он переходит (по фронту Clk) в новое состояние. В какое именно состояние перейдет автомат, определяется входным сигналом, значение которого должно быть стабильным в течение определенного времени перед следующим фронтом Clk (для выполнения корректного перехода, определенного алгоритмом работы автомата).

Входные события можно классифицировать как инициирующие и прерывающие (аварийные). Иницирующие события запускают переход автомата в новое состояние и выдачу соответствующих выходных сигналов. Прерывающее событие прерывает переход у автомата Мили или прерывает задержку в состоянии автомата Мура. Иначе говоря, прерывающее событие меняет значение выходного сигнала до его «стандартного» окончания. Но при этом некоторые инициирующие события можно рассматривать как входные воздействия, например включение питания или запуск какого-либо процесса в управляемой системе.

Если у структурного автомата часть входных сигналов рассматривается как входные воздействия (которые опрашиваются), а часть входных сигналов рассматривается как события, то такой автомат принято называть смешанным. При выборе типа управляющего автомата в системах логического управления следует учитывать оба указанных параметра.

Любой активный автомат, где есть прерывающее событие, по сути, является смешанным, а кроме того, определенный входной сигнал может одновременно рассматриваться как входное воздействие или событием, что вместе с тем не нарушает приведенной классификации.

Если рассматривать автоматы по классификации по выходным сигналам, то автоматы Мура, как правило, рассматриваются как активные или смешанные. Такие модели составляют большую часть управляющих автоматов в системах логического управления, т.к. достаточно просто установить соответствие между техническими состояниями управляемого объекта и набором управляющих сигналов в состояниях управляющего автомата Мура.

Пассивные автоматы Мура рассматриваются достаточно редко.

Таймаут для активного автомата Мура соответствует задержке нахождения определенном техническом состоянии. Активный автомат Мура считывает (опрашивает) входной сигнал в момент окончания фронта первого синхросигнала после окончания таймаута  $t_{0i}$  текущего состояния и «мгновенно» переходит в новое состояние. Выходные сигналы автомата Мура связаны с текущим состоянием автомата. Для каждого выходного сигнала  $u_j$  определяется задержка его появления (изменения) от момента перехода в текущее состояние. После окончания таймаута выходной сигнал может обнуляться или продолжаться в зависимости от особенностей алгоритма работы автомата Мура.

Для реализации автоматной модели Мура с единственной временной переменной на языке описания аппаратуры применяется дополнительный счетчик count, который используется для подсчета числа автоматных тактов, в течение которых автомат реализует определенные временные параметры. Когда временной автомат выполняет переход в новое состояние, значение count, сбрасывается в 0. В двухпроцессном шаблоне на языке VHDL назначения нового состояния и нового значения счетчика происходят в одном процессе, связанном с синхронизацией и установкой автомата в начальное состояние.

Таймаут  $t_{0i}$  реализуется многократным переходом из состояния в это самое состояние, при этом число переходов определяется числом автоматных тактов таймаута. Значение счетчика сравнивается с  $(t_{0i}-1)$ , поскольку при переходе в состояние  $a_i$  автомат один такт находится в нем и, чтобы таймаут был точно равным  $t_{0i}$  тактов, необходимо еще  $(t_{0i}-1)$  тактов повторения. В темпоральном графе переходов таймауты состояний указываются внутри окружностей обозначения состояний и реализуются с помощью петель, условиями для которых являются проверки значения сигнала счетчика автоматных тактов, но эти петли на графе, как и сигнал Clk, не помечаются.

Задержки выходных сигналов в текущем  $i$ -ом состоянии  $td_{ij}$  определяются в автоматных тактах от момента перехода автомата в соответствующее состояние для каждого выходного сигнала  $u_j$  и реализуется путем анализа значений счетчика, который определяет задержку данного сигнала в автоматных тактах. На темпоральном графе переходов выходная задержка указывается в скобках рядом с каждым выходным сигналом, а в VHDL-коде реализуется оператором условного назначения сигнала вне процесса.

Модель активного автомата Мура при  $t_{0i}=1$  и  $td_{ij}=0$  совпадает с моделью традиционного микропрограммного автомата.

Моделі временних управляючих автоматів в системах логічного управління, які реалізуються в hardware, характеризуються достаточним різноманітністю і мають достатньо багато класифікацій. По способу формування вихідних сигналів, автомати поділяються на моделі Мура і Мілі [10]. По типу і способу обробки входних сигналів автомати класифікуються на опрацюючі входні сигнали і події, а події в свою чергу класифікуються на зовнішні по відношенню до управляємої системи і внутрішні, при цьому зовнішні події також мають багато класифікацій [1, 8]. По способу кодування станів автомати діляться на ряд класів, для яких апаратні витрати і швидкість суттєво різняться [7]. По способу обробки временних параметрів для временних автоматів вони класифікуються на автомати з таймаутами, временними обмеженнями, временними затримками і їх комбінаціями [3, 5]. По способу реалізації переходів і отримання вихідних сигналів FSM класифікуються на регулярні (regular), временні (timed) і рекурсивні (recursive) [9, 10]. Таке різноманітність класифікацій управляючих автоматів в hardware обумовлено широкою областю їх застосування і різноманітністю розв'язуваних завдань. Спосіб класифікації моделей FSM є визначальним при побудові шаблонів (patterns) HDL-моделей временних управляючих автоматів.

В даній роботі авторами запропонована класифікація автоматів по способу отримання вихідних сигналів на моделі Мура і Мілі, по способу обробки входних сигналів на активні і пасивні моделі і класифікація подій по способу їх обробки на ініціюючі і перериваючі. Данна класифікація дозволила побудувати VHDL-шаблони моделей временних управляючих автоматів для розв'язання різноманітних завдань в системах логічного управління. Моделювання, синтез і імплементація в FPGA підтвердили належність розроблених шаблонів до синтезованого підмножини VHDL і дотримання временних параметрів, заданих специфікаціями.

Напрямок подальших досліджень може бути пов'язано з розробкою і аналізом HDL-моделей на мові Verilog і розробкою HDL-шаблонів для рекурсивних автоматів.

#### Список літератури

4. Shalyto A.A. Software Automation Design: Algorithmization and Programming of Problems of Logical Control / A.A. Shalyto // Journal of Computer and System Sciences International. – 2000. – Vol. 39, No. 6. – P. 899-916.

5. Zhigulin M. FSM-Based Test Derivation Strategies for Systems with Time-Outs / M. Zhigulin, N. Yevtushenko, S. Maag, A.R. Cavalli // Proceedings of the 11th International Conference on Quality Software (QSIC 2011), Madrid, 2011. – P. 141-149.

6. Gromov M. Testing Components of Interacting Timed Finite State Machines / M. Gromov, A. Tvardovskii, N. Yevtushenko // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'16), October 14-17, Yerevan, Armenia, 2016. – P. 193-196.

7. Tvardovskii A.S., Yevtushenko N.V., Gromov M.L. Minimizing Finite State Machines with time guards and timeouts // Proc. ISP RAS. – 2017. – vol. 29. – issue 4, – P. 139-154.

8. Bresolin D. Minimizing Deterministic Timed Finite State Machines / D. Bresolin, A. Tvardovskii, N. Yevtushenko, T. Villa, M. Gromov // In 14th IFAC Workshop on Discrete Event Systems WODES 2018. – IFAC-PapersOnLine, 2018. – Vol. 51. – issue 7. – P. 486-492.

9. Solov'ev V.V. Structural models of finite-state machines for their implementation on programmable logic devices and systems on chip / A.S.Klimowicz, V.V. Solov'ev // Journal of Computer and System Sciences International. – 2015. – V. 54. – № 2. – P. 230-242.

10. Wagner G. An abstract state machine semantics for discrete event simulation // Proceedings of the 2017 Winter Simulation Conference (WSC), 3-6 Dec. 2017, Las Vegas, USA – 12 p. [Electronic resource] / IEEE Xplore Digital Library – Access mode: www / URL: <https://ieeexplore.ieee.org/document/8247830>.

11. Pedroni, V. A. Finite state machines in hardware: theory and design (with VHDL and SystemVerilog) / Volnei A. Pedroni. – Cambridge, MA: MIT Press., 2013. – 338 p.

12. Shkil A. Design of real-time logic control system on FPGA / M. Miroschnyk, A. Shkil, E. Kulak, D. Rakhlis, I. Filippenko, M. Hoha, M. Malakhov, V. Serhiienko // Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS'19), September 13-16, Batumi, Georgia, 2019. – P. 488-491.

13. Baranov S. Logic and System Design of Digital Systems / S. Baranov. – Tallinn: TUT Press, 2008. – 267 p.