UDC 004.93

TEMPORAL EVENTS PROCESSING MODELS IN FINITE STATE MACHINES

Miroshnyk M. A. – Doctor of Science, Professor, Professor of the Department of Theoretical and Applied Systems Engineering, V. N. Karazin, Kharkiv National University, Kharkiv, Ukraine.

Shmatkov S. I. – Doctor of Science, Professor, Head of the Department of Theoretical and Applied Systems Engineering, V. N. Karazin, Kharkiv National University, Kharkiv, Ukraine.

Shkil O. S. – PhD, Associate Professor of the Department of Computer Engineering Design Automation, Kharkiv National University of Radioelectronics, Kharkiv, Ukraine.

Miroshnyk A. M. – Assistant of the Department of Computer Engineering Design Automation, Kharkiv National University of Radioelectronics, Kharkiv, Ukraine.

Pshenychnyi K. Y. – Postgraduate student of the Department of Computer Engineering Design Automation, Kharkiv National University of Radioelectronics, Kharkiv, Ukraine.

ABSTRACT

Context. The issue of a synthesizable finite state machine with temporal events processing using hardware description language pattern. The object of this study is external event processing in real-time systems.

Objective. The goal of this work is to introduce methods to express external temporal events on finite state machine state diagrams and corresponding HDL patterns of such events processing in control systems.

Method. The classification of external events in real-time systems is analyzed. A device class that changes its internal state depending on the temporal external events is introduced. A method to express these events on the temporal state diagram is introduced. Possible model behavior scenarios based on the external event duration are analyzed. A Verilog HDL external event processing pattern is introduced. The efficiency of the proposed model is proved by developing, verifying, and synthesis of a power-saving module in Xilinx ISE. The results and testing showed the model's correctness.

Results. External temporal events processing methods in real-time device models are proposed. The corresponding HDL pattern for the proposed model implementation is presented.

Conclusions. The real-time systems with external temporal events automated synthesis problem has been solved. To solve this problem, a finite state machine model-based device using the Verilog language was developed and tested. The scientific novelty lies in the introduction a method to express temporal events on the state diagram of the finite state machine as well as in a HDL when implementing the proposed model on CPLD and FPGA.

KEYWORDS: FSM pattern, HDL, real time devices, temporal events, electronic design automation.

ABBREVIATIONS

FSM is a finite state machine;
CAD is a computer aided design;
RTL is a register transfer level;
HDL is a hardware description language.

NOMENCLATURE

$X = \{X_C, X_E\}$ is a set of input variables;

X_C is a set of input signals from the control object;

X_E is a set of external events;

$Y = \{Y_C, Y_F\}$ is a set of output variables;

Y_c is a set of control functions;

Y_f is a set of initial functions;

Z is a set of internal variables that determine the encoding of FSM states;

f is a transition function;

g is an output function;

Z_0 is a initial FSM state;

$T_c = \{t_{c1}, t_{c2....} t_{cp}\}$ is a set of timed variables for timing constraints on each arc of the state diagram;

p is a number of arcs in the state diagram;

$t_{ci} = \{1, k\}$ is a maximum number of constraints on transitions to the i-th node of the state diagram in the event processing mode;

$T_{to} = \{t_{to1}, t_{to2} ... t_{ton}\}$ is a set of timed variables for timing constraints for timeout of each FSM state;

$t_{toi} = \{1, n\}$ is a timeout for each state;

n is a number of FSM states;

$T_d = \{t_{d1}, t_{d2} ... t_{dm}\}$ is a set delays for implementing output signal;

m is a number of output variables;

$t_{dm} = \{1, l\}$ is a range of possible delay values for each output signal;

l is a maximum number of clock cycles for implementing output signals in the indicated state of the FSM.

INTRODUCTION

Logic control systems represent a significant node any digital system. Such systems use the binary alphabet to determine the behavior of a control unit. The FSM pattern is known to be a widespread model for such systems. FSM itself is mathematical abstraction that can be represented in various ways such as state-transition table, state or flow-chart diagram, etc. When using the FSM pattern, it is essential to understand that the target logic unit behavior depends on the events happening in the external environment.

The clock cycles determine the machine time in which FSMs operate. However, the real-time devices work in the

DOI 10.15588/1607-3274-2023-4-5

metric time. In other words, such devices state depends both on the input signals and the time during which these signals are processed [2]. Thereby there are arises a requirement to express the metric time in terms of clock cycles because transitions between the FSM states directly depend on the timing aspect. It's also required to express the timing constraints on the state diagram as a starting modeling point.

HDLs has proven their indispensable role in both modern hardware development and verification processes. The high level of abstraction provided by language construct allows to implement algorithms of any difficulty. Modern CAD systems provide a range of tools to examine design before actual circuit implementation allowing to check the model correctness.

The automata-based description style is a way to structure HDL description when implementing a control logic device. It's the automata pattern that defines the behavior of the target device including timing parameters such as delays and duration requirements.

Thus, there exists a task of developing a unified HDL pattern for real time devices with external temporal events that can be used for the device implementation on PLD hardware platform (FPGA, CPLD).

The object of study is automatized real-time digital control logic devices development process.

The subject of study is events model and their processing in FSM-based real-time devices.

1 PROBLEM STATEMENT

Suppose given general temporal control unit model: $Y(t) = g(X(t), Z(t), T)$, $Z(t+1) = f(X(t), Z(t), T)$. Where $T = \{t_c, t_o, t_d\}$ represents automata time parameters: t_c – time constraints, t_o – output timeouts, t_d – input delays.

For given input constraints $t_c(X(i)) = [c1]$ that represents external event the problem of processing these events in real time control logic device models can be presented.

The main goal of this research is to study the usage of external temporal events in real time device. Such events have a requirement of a minimal duration required for a device response. Also, this paper addresses the following problems:

– define a class of devices that depend on the external temporal events;

– define a way to represent this event class on the FSM state diagram;

– simulate possible scenarios of such event processing;

– introduce HDL patterns that would express temporal events processing.

The object of study is real-time devices control algorithms.

The subject of study is real-time control device automata patterns.

2 REVIEW OF THE LITERATURE

In [1] any computing device is represented as a combination of two parts –operational unit and computing unit. This form of representation is based on the following principles:

– any operation can be represented as a sequence of basic logical and arithmetic operation upon input data;

– logical condition define the sequence of the operations;

– microprogram is representation of an algorithms in terms of microoperations and logical conditions;

– microprogram defines the structure of a device and the way it operates in time.

Operational unit stores words of information, performs operations upon them and calculates logical conditions required for algorithm. Operation unit us defined by a range of finite sets:

$Y = \{y_1, y_m\}$ defines microoperations;

$X = \{x_1, x_l\}$ defines logical conditions that are required during the algorithm flow;

$D = \{d_1, d_h\}$ represents operands;

$R = \{r_1, r_q\}$ represents operation results;

$S = \{s_1, s_n\}$ – internal words that represent information during computations.

The control unit generates a sequence of control signals based on logical conditions. This way the control unit defines the order of the operations.

The timed FSM as way to describe real-time devices was introduced in [2, 3]. In these works, the state diagram is extended with a finite state of real value timers. Each timer is reset to zero during a transition and increased with each FSM cycle. Each transition has an associated clock constraint. This means that transition can only be executed when all timer values satisfy this restriction. These papers introduce a generalized model of temporal control unit: $Y(t) = g(X(t), Z(t), T)$, $Z(t+1) = f(X(t), Z(t), T)$. Here X is the set of input signals, Z – the set of internal automata states, Y – the set of output signals, t – the machine time defined by clock cycles, d – the output function, t – the transition function. $T = \{t_c, t_o, t_d\}$ is the set of automata time parameters: t_c – time constraints, t_o – output timeouts, t_d – input delays.

In [2] all state machines are classified into three categories: regular, timed and recursive. Here the timed FSM are the FSM which have at least one time-dependent transition. For each machine category a general HDL pattern is proposed. As in [3] state loops and additional variable is used to implement the delays between state transitions.

In [4, 5] timed FSM that consider state timeouts and output signals delays. In the proposed model FSM makes a transition into a determined state in case no input signals were triggered during a timeout. Testing methods for TFSM were considered as well.

The method for implementing models of FSM in the VHDL language was proposed in [7, 8]. To implement delays in the states of the Moore FSM, it is proposed to use loops in states and a special variable count, which

DOI 10.15588/1607-3274-2023-4-5

decreases by 1 in each FSM cycle, which corresponds to a sync pulse.

Event based timed automata description and minimization method are described in [9, 10].

The method for constructing a HDL description for real-time systems, which can be synthesized into a programmable logic device, was developed in the [13, 14]. Models of real-time systems, such as a state diagram, a timed FSM with many timers, a timed FSM with one timer, an extended timed FSM, a timed FSM with timeouts and timing constraints were considered. Based on these models, a complete structural model of timed Moore FSM was proposed.

The issue of finding homing sequences for finite state machines and their length is addressed in [11]. Lower/Upper-bound parametric timed automata without invariants is described in [12].

In [16] timed systems in which some timing features are unknown parameters are addressed. The Upper-bound Parametric Timed Automata (U-PTAs), one of the simplest extensions of timed automata with parameters, in which parameters are only used as clock upper bounds are addressed.

In [17] logical discrete event systems modeling is addressed. It also discusses the diagnosability and opacity in the context of partially observed systems.

The issue of modeling and verification of digital discrete event systems is discussed in [18]. This work introduces the connection between Event-B methodology and automata modeling. This paper is important for verification of the automata based models.

In [19] a new classification of system events is during the proposed. This classification is used during the software and hardware specification development process. There are three classes of external events:

– business event – an action by a human user that stimulates a dialog with the software or hardware, as when the user presses a button;

– signal event – such event is registered when the system receives a control signal, data reading, or interrupt from an external hardware device or another software system;

– temporal event – a time-triggered event, as when the computer's clock reaches a specified time or when a preset duration has passed since a previous event (as in a system that logs a sensor's temperature reading every 10 seconds).

None of the works describes automata events with a duration longer than 1 automata cycle. In this paper, the FSM models with external temporal events are considered.

3 MATERIALS AND METHODS

The temporal state diagram is a visual representation of the real-time automata. This state diagram is extended with a timer used for state delays. The timer is used for keeping the automat in a certain state during a fixed amount of clock cycles. An example of timed FSM is shown on Fig. 1.

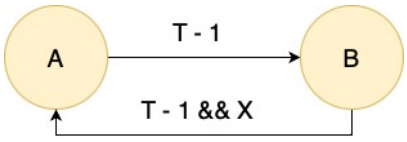

Figure 1 – Timed FSM example

Here the *AB* transition is a regular timed transition during which the automata remains in the state *A* during *T* clock cycles. The condition is written in the form $T - 1$ because the internal timer will run from $t = 0$ to $t = T - 1$. Here the *BA* transition depends on the auxiliary timer state and the input signal *X* (conditionally timed transition). Thereby the FSM will remain in state *B* at least during $T - 1$ clock cycles. From the HDL point of view, the time-related transitions are implemented via staying in a particular state, i.e., state loop. This transition type is called conditionally-timed.

Fig. 2 elaborates on both timed and conditionally timed transitions for the FSM from the Fig. 1 using an explicit timer value in the condition description.

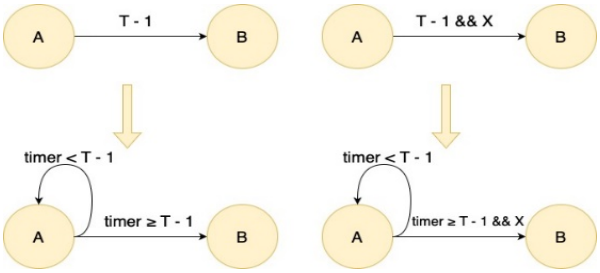

Figure 2 – Timed transitions

As it was already said there exists a class of devices that are sensible to external events with a certain duration. Event time parameters are considered when modeling such devices. For example, pushing the button for certain time turn-on a device; cooling must be enabled in a case a certain temperature level is maintained in a room; a system must transfer to the emergency mode in case the operator ignores error messages for 5 minutes. The mentioned device class must be described using the existing automata-based methods.

The event time constraints is a value $t_c(X(i)) = [c1]$ that represents the minimal duration of a signal. In the button example the $t = 3$ seconds. This constraint must be expressed on the temporal state diagram as transition. Such transition must explicitly compare the timer value with the $t_c(X(i))$ value.

Such transition is shown on Fig. 3. Like with regular timed transition the condition is based upon auxiliary timer value, but in this case the timer must have a minimal value for a state transition.

DOI 10.15588/1607-3274-2023-4-5

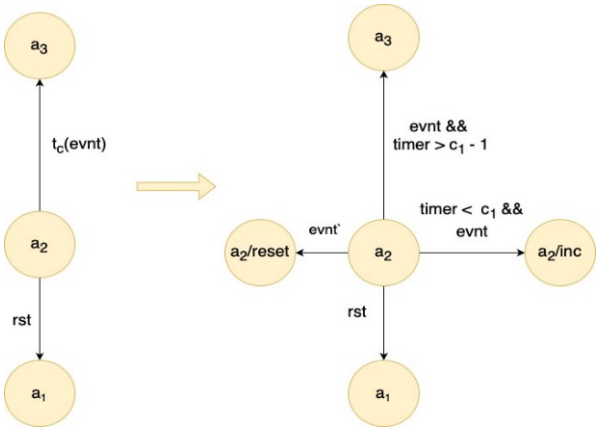

Figure 3 – Event with time constraint transition

For example, for the successful state transition between a_1/a_2 signal *evnt* must be active high and the timer value is greater than $t_c(evnt)$.

Internally the a_2 state consists of 2 states: *a*2/*reset* and *a*2/*inc*. In case the signal *evnt* is high, but the timer value is less than $t_c(evnt)$ the automata transfers into the and *a*2/*inc* substate where the timer is incremented. In case the signal *evnt* is low and the timer value is less that $t_c(evnt)$ – the automate goes into the substate *a*2/*reset* where the timer is reset to 0. It's essential that automata has a timer-independent transition to avoid livelocks. In this example this is the a_2/a_1 transition activated by signal *rst*.

There are four possible scenarios of automata behavior depending on the nature of external events. To illustrate these examples let's consider an automata transition that occur after signal *evnt* maintains high logical level during at least 4 clock cycles. The first scenario describes a situation when the external event lasts exactly 4 clock cycles. In this case the automata transfers into the next state and the internal counter is reset. Figure 4 shows a waveform of this scenario – *evnt* maintains high logical level during 4 clock cycles (the minimal amount of clock cycles required for the transition) and the automat transfers to state *a*3. The calculation points are marked with a red dotted line.

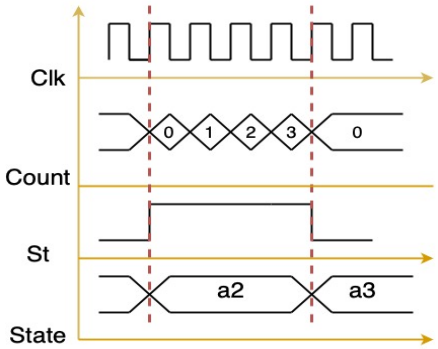

Figure 3 – Exact match event waveform

The next situation describes a case when the external event last longer than required. It worth noting that for the observed devices a longer event is not critical, because the minimal event duration condition is satisfied. Like in the previous case the automata will transfer to the next state and the internal timer will be reset. Corresponding waveform is shown on Figure 4. Signal *evnt* remains high during 5 clock cycles that is 1 clock cycle longer than required. After 4 cycles the automata transfers to the next state and the timer is reset. The event calculation points are marked with green dotted line, the automata transfer is marked with red dotted line.

The last scenario is when the event does not satisfy the timing requirements, i.e., lasts less than required. In this case the automata preserves its state with the timer being reset. Resetting the timer is an important operation, because the timer must be ready for the next event calculation. The waveform for this scenario is shown on Figure 5.

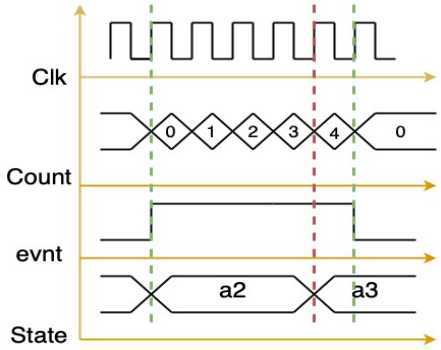

Figure 4 – Longer event waveform

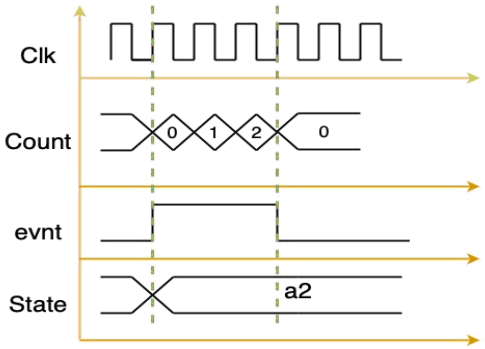

Figure 5 – Shorter event waveform

When describing temporal automata using state loops the deadlock situation must be addressed. The deadlock may occur in case the desired event does not occur at all. Thus, there must be a way to reset the automata into a predefined state. Typically a digital device has a reset signal with a higher priority than other signals. The priority is important when describing automata using hardware description languages like Verilog, VHDL. Figure 6 shows a waveform for the situation when both *evnt* and *rst* signals remain high, but the machine goes to the state a1, because *rst* has a higher priority.

DOI 10.15588/1607-3274-2023-4-5

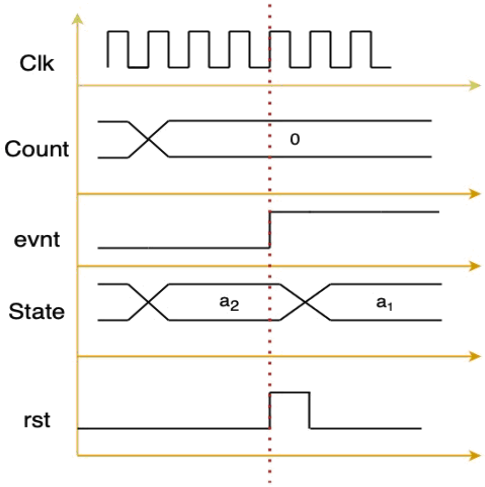

Figure 6 – A sync reset waveform

4 EXPERIMENTS

To illustrate the proposed event-based timed FSM model the power save module will be used. This module is usually used in power critical systems to decrease the energy usage whenever the system does not operate for a certain period.

The module operates in two modes: bypass and power saving. The input alphabet consists of the following binary signals X = {*onn, evnt*}, where *onn* – signal to enable power saving algorithm, *evnt* – signal notifying that an external event occurred, and the system must leave the power saving mode. The output binary signals set is Y = {*save*}, where *save* represent the power saving mode enter. The module interface and its connection with other system nodes are shown on Fig. 7.

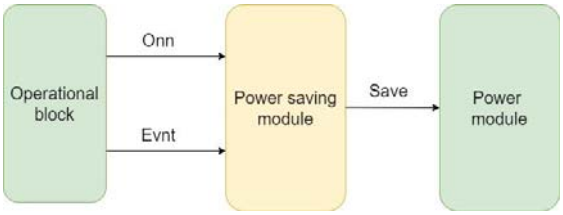

Figure 7 – Power saving module interface

The next step is to define the FSM states and timing constraints. The states set is the following:

– state a1 – the module is working in bypass mode, i.e. no external events are monitored;

– state a2 – power saving mode computation – in case no events occur during a fixed time period the FSM moves into the a3 state;

– state a3 – power saving mode.

The algorithm of the save power module is quite straightforward. Whenever this block is enabled by the *onn* signal it starts the *evnt* input signal monitoring. In case the *evnt* signal is not asserted during a fixed amount of time – the system must enter the power saving mode (output signal *save* is set high). The system exits the saving mode in case the *evnt* signal is asserted or the module is disabled via *onn* input signal with subsequent *save* signal de-assertion.

Thus, for the power saving module temporal state diagram of timed Moore FSM (Figure 8) with a proper HDL implementation can be built.

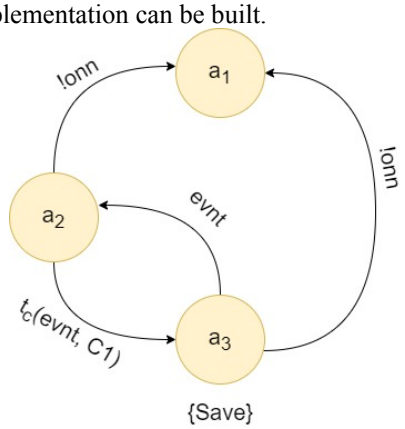

Figure 8 – Power saving module Moore timed FSM state diagram

Figure 9 represents FSM next-state Verilog process description.

Figure 10 shows the waveform of the module usage. Here t_c equals 5 clock cycles. The event calculation period is marked by red lines; blue lines denote signals change (*evnt* and *onn*) making FSM go into the event monitoring or bypass state.

CAD identified an FSM with 3 states for both XC3S500E-5fg320 and CPLD XC9572XL-53C44 chips. The state encoding is equal for both devices which is: a1=”00”, a2=”01”, a3=11”.

```
always @(state, evnt, onn, count)
  case (state)
    a1: if (onn) begin
          next_state = a2;
        end
       else begin
          next_state = a1;
       end

    a2: if (!onn) begin
          next_state = a1;
        end
        else if (count >= C1 - 1 && !evnt) begin
          next_state = a3;
          next_count = 3'd0;
        end
        else if (!evnt) begin
          next_state = a2;
          next_count = count + 1'b1;
        end
        else begin
          next_state = a2;
          next_count = 3'd0;
        end

    a3: if (!onn) begin
          next_state = a1;
        end
        else if (evnt) begin
          next_state = a2;
        end else begin
          next_state = a3;
        end

    default: begin
      next_state = a1;
      next_count = 3'd0;
    end
  endcase
```

Figure 9 – FSM next state process Verilog description

DOI 10.15588/1607-3274-2023-4-5

5 RESULTS

Xilinx ISE 14.7 CAD system was used for the design verification, implementation, and synthesis. Behavioral and post-implementation simulations for initial description were performed on the CPLD XC9572XL-10-TQ100 (Post-Fit Simulation) and on the FPGA XC3S500E-5fg320 (Post-Place & Route Simulation).

Figures 10 and 11 shows the post synthesis waveform for XC3S500E-5fg320 and XC9572XL-53C44 correspondingly.

Switching delay is 0 ns. Single short-term pulses do not occur. Thus, when implementing the device on FPGA and CPLD, its operation must comply with the original description (specification).

The expected minimum number of triggers is 9: 3 triggers for encoding 7 states, 6 triggers for the counter (since the maximum timeout is 40 clock cycles). RTL schematic report for both chips confirmed this. (since the maximum timeout is 40 clock cycles). RTL schematic report for both chips confirmed this.

Latch triggers are absent in the report. To implement functions of transitions and outputs, combinational circuits were synthesized. 52 combinational elements were synthesized on the XC9572XL-10-TQ100, and 21 combinational elements – on the XC3S500E-5fg320. Table 1 shows some totals from the synthesis report.

For FPGA: minimum clock period: 4.153 ns (Maximum Frequency: 240.790 MHz).

For CPLD: minimum clock period: 6.300 ns (Maximum Frequency: 158.730 MHz).

Table 1 – FPGA and CPLD based FSM synthesis results

PLD	Flip-Flops	Latches	BELS	Slices / LUTs
FPGA XC3S500E5fg320	8	0	21	11/21
CPLD XC9572XL-TQ100	8	0	52	–

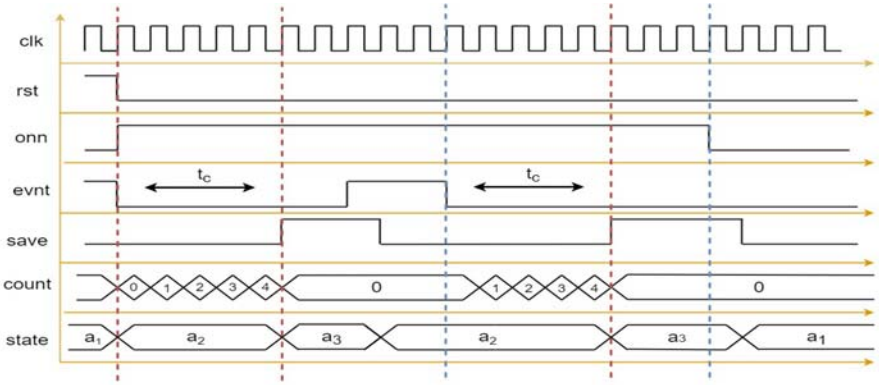

Figure 10 – Timing diagram of Moor FSM for power saving module

Figure 11 –Post-Place & Route Simulation of XC3S500E-5fg320

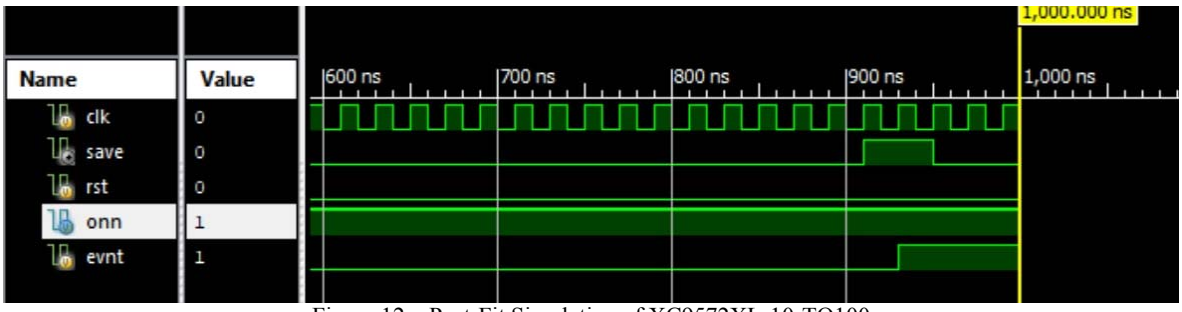

Figure 12 – Post-Fit Simulation of XC9572XL-10-TQ100

DOI 10.15588/1607-3274-2023-4-5

6 DISCUSSION

Moore FSM models with timing delays represent a traditional way for describing logic control systems. Timed FSM models with timeouts, constraints and output delays are used during control systems and telecommunication protocol testing [5–10]. In [13–15] a hardware implementation of such FSM models using hardware description languages was introduced. In [4] VHDL patterns for timed Moore control systems were presented. These patterns consider event processing external to the control system. However, no HDL patterns for processing external events that must satisfy certain timing constrains were mentioned.

In this research a general HDL pattern for processing temporal events was introduced. The model synthesizability was proven by implementing power saving model in Xilinx IDE on two chips – FPGA and CPLD. The results shown different hardware usage results. At the same time the behavioral simulation confirmed both devices satisfy specification.

Further works might be related to the verification of such HDL models, especially using assertion-based methods [2].

CONCLUSIONS

The issue of real-time devices modeling using the finite state machine pattern and hardware description languages was solved. The classification of events as models of finite state automaton interaction with the external environment is analyzed. Temporal events and a class of devices which functionality depends on such factors is are discussed. It is shown that such events are widespread in various digital devices and real-time systems. A new type of FSM transition is proposed for the temporal transitions state diagram as a digital systems design canonical model. The considered models are illustrated by simulations and timing diagrams analysis. The work solves the problem of designing time-controlled automata in real-time logic control systems. The issue of logical systems with external events processing that should last a certain time is considered.

An algorithm of a power saving unit was analyzed. Based on this algorithm, the temporal Moore state diagram was developed. The temporal state diagram was used to develop a three-process Verilog-model of the timed Moore FSM.

Verification, synthesis, and implementation of the developed Verilog model using Xilinx ISE environment were performed. Synthesis and simulation before and after implementation were performed for CPLD XC9572XL-10-TQ100 and FPGA XC3S500E-5fg320. Synthesis and simulation results of the circuit after implementation confirm the operability and correctness of the developed Verilog model.

The scientific novelty of the work lies in temporal state diagram improvement. This allowed to expand the class of event based real-time logical devices represented by FSM.

The practical significance of the obtained results lies in the following:

– investigate a class of real time control logic devices with corresponding FSM models and HDL description is introduced;

– the possibility of the proposed model is illustrated with examples that show various correlation between event duration and the time during which control FSM stays in a certain state;

– the theoretical basis is illustrated with the control unit for power saving device implementation on FPGA using Verilog HDL.

Prospects for further research are to use the proposed theoretical basis and results for event-based devices diagnostics systems [15].

ACKNOWLEDGEMENTS

The work was carried out within the framework of the state budget research topic of the Kharkov National University of Radioelectronics “Smart cyber university – Cloud-Mobile services for managing scientific and educational processes l” (state registration 0017U002524, order of the Ministry of Education of Ukraine) on the basis of the Design automation department of KNURE.

And out within the framework of the state budget research topic of the V. N. Karazin, Kharkiv National University “Modeling of information processes in complex and distributed systems” (state registration number 0121U109183, order of the Ministry of Education of Ukraine) on the basis of the of Theoretical and Applied Systems Engineering department of KNUK.

REFERENCES

1. Baranov S. Logic and System Design of Digital Systems. – Tallinn, TUT Press, 2008, 267 p. doi.org/10.1016/B978-0-7506-8397-5.00005-2.
2. Pedroni V. A. Finite state machines in hardware: theory and design (with VHDL and SystemVerilog). Cambridge, MA: MIT Press, 2013, 338 p. doi.org/10.7551/mitpress/9657.001.0001
3. Miroschnyk M., Shkil A., Kulak E., Rakhlis D., Filippenko I., Hoha M., Malakhov M., Serhiienko V. Design of real-time logic control system on FPGA, *Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS’19), September 13–16.* Batumi, Georgia, 2019, pp. 488–491. doi.org/10.1109/ewdts.2019.8884387
4. Shalyto A. A. Software Automation Design: Algorithmization and Programming of Problems of Logical Control, *Journal of Computer and System Sciences International*, 2000, Vol. 39, No. 6, pp. 899–916. doi.org/10.1023/a:1012392927006
5. Alur R., Dill D. L. A theory of timed automata, *Theoretical Computer Science*, 1994, Vol. 126, No. 2, pp. 183–235. doi.org/10.1016/0304-3975(94)90010-8
6. Zhigulin M., Yevtushenko N., Maag S., Cavalli A. R. FSM-Based Test Derivation Strategies for Systems with Time-Outs, *Proceedings of the 11th International Conference on Quality Software (QSIC 2011)*. Madrid, 2011, pp. 141–149. doi.org/10.1109/qsic.2011.30
7. Solov’ev V. V., Klimowicz A. S., Structural models of finite-state machines for their implementation on

DOI 10.15588/1607-3274-2023-4-5

programmable logic devices and systems on chip, *Journal of Computer and Systems Sciences International*, 2015, Vol. 54, № 2, pp. 230–242. doi.org/10.1134/s1064230715010074
8. Shkil A. S., Miroschnyk M. A., Kulak E. N., Rakhlis D. Y., Miroshnyk A. M., Malahov N. V. Design timed FSM with VHDL Moore pattern, *Radio Electronics, Computer Science, Control*, 2020, № 2(53), pp. 137–148. /doi.org/10.15588/1607-3274-2020-2-14
9. Bresolin D., Tvardovskii A., Yevtushenko N., Villa T., Gromov M. Minimizing Deterministic Timed Finite State Machines, *In 14th IFAC Workshop on Discrete Event Systems WODES 2018. – IFAC-PapersOnLine,* 2018, Vol. 51, Issue 7, pp. 486–492. /doi.org/10.1016/j.ifacol.2018.06.344
10. Bresolin D., El-Fakih K., Villa T., Yevtushenko N. Equivalence checking and intersection of deterministic timed finite state machines, *Formal Methods in System Design*, 2022, № 7, pp. 1–26. doi.org/10.1007/s10703-022-00396-6
11. Tvardovskii A. S., Yevtushenko N. V. Deriving homing sequences for finite state machines with timed guard, *Automatic Control and Computer Sciences,* 2021, Vol. 55, № 7, pp. 738–750. doi.org/10.3103/s0146411621070154
12. André Etienne, Lime Didier, Ramparison Mathias TCTL Model Checking Lower/Upper-Bound Parametric Timed Automata Without Invariants, *Proc.International Conference Formal Modeling and Analysis of Timed Systems FORMATS 2018, September 4–6*. Biejing, China, 2018, pp. 37–52. doi.org/10.1007/978-3-030-00151-3_3
13. Wagner G. An abstract state machine semantics for discrete event simulation, *Proc. 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. /doi.org/10.1109/wsc.2017.8247830
14. Shkil A., Miroschnyk M., Kulak E., Rakhlis D., Filippenko I., Malakhov M. Hardware implementation of timed logical control FSM, *Proc. of 2020 IEEE East-West Design & Test Symposium (EWDTS'20), Sept. 4–7*. Varna, Bulgaria, 2020.– 6 p. [Electronic resource] / IEEE Xplore Digital Library – Access mode: www / URL: https://ieeexplore.ieee.org/document/ 9225129. /doi.org/10.1109/ewdts50664.2020.9225129
15. Lamperti G. Zanella M., Zhao Xiangfu Introduction to Diagnosis of Active Systems. Springer, 2018, 353 p. /doi.org/10.1007/978-3-319-92733-6
16. André Etienne, Lime Didier, Ramparison Mathias TCTL Model Checking Lower/Upper-Bound Parametric Timed Automata Without Invariants, *Proc.International Conference Formal Modeling and Analysis of Timed Systems FORMATS 2018, September 4–6*. Biejing, China, 2018, pp. 37–52. doi.org/10.1007/978-3-030-00151-3_3
17. Stéphane Lafortune. Discrete Event Systems: Modeling, Observation, and Control, *Annual Review of Control, Robotics, and Autonomous Systems*, 2019, No. 2:1, pp. 141–159. doi.org/10.1146/annurev-control-053018-023659
18. Sabah Al-Fedaghi. Modeling Physical / Digital Systems: Formal Event-B vs. Diagrammatic Thinging Machine, *International Journal of Computer Science and Network Security*, 2020, No. 20 (4), pp. 208–220. hal-02614504 , version 1 (20-05-2020)
19. Karl Wiegers, Beatty Joy Software Requirements (Developer Best Practices) 3rd Edition, Developer Best Practices, 2013, 672 p. /doi.org/10.1109/9781118156629.ch3

Received 15.08.2023.
Accepted 04.11.2023.

УДК 004.93

МОДЕЛІ ТЕМПОРАЛЬНИХ ПОДІЙ У КІНЦЕВИХ АВТОМАТАХ

Мірошник М. А. – д-р техн. наук, професор, професор кафедри теоретичної та прикладної системотехніки Харківського національного університету імені В. Н. Каразіна, Харків, Україна.

Шматков С. І. – д-р техн. наук, професор, завідувач кафедри теоретичної та прикладної системотехніки Харківського національного університету імені В. Н. Каразіна, Харків, Україна.

Шкіль О. С. – канд. техн. наук, доцент, професор кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна.

Мірошник А. М. – асистент кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна.

Пшеничний К. Ю. – аспірант кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу розробки шаблонів кінцевих автоматів з обробкою зовнішніх темпоральних подій з використанням мов опису апаратури. Об'єктом роботи є питання моделювання зовнішніх подій у системах реального часу.

Мета роботи. Метою роботи є представити способи вираження темпоральних подій у на графі переходів кінцевого автомата, а також відповідні HDL шаблони обробки таких подій у системах управління.

Метод. Проаналізовано класифікацію зовнішніх подій у системах реального часу. Виділено клас пристроїв, у яких зміна стану відбувається внаслідок настання зовнішніх подій, що подовжені у часі (темпоральні події). Запропоновано спосіб вираження такого роду подій на темпоральному графі переходів кінцевого автомата. Проаналізовано різні сценарії поведінки запропонованої автоматної моделі в залежності від тривалості зовнішньої події. Розроблено HDL шаблони на мові опису апаратури Verilog для імплементації обробки темпоральних подій. Працездатність запропонованих методів доведено на прикладі розробки, верифікації та синтезу модуля збереження енергії на FPGA та CPLD у системі автоматизованого проектування Xilinx ISE. Отримані результати автоматизованого синтезу довели правильність запропонованої методології.

Результати. Запропоновано методи обробки зовнішніх темпоральних подій у моделях пристроїв реального часу. Представлено відповідні шаблони мові опису апаратури Verilog для імплементації запропонованої моделі.

Висновки. Вирішено задачу автоматизованого синтезу систем реального часу з зовнішніми темпоральними подіями. Для вирішення цієї проблеми були розроблені та протестовано модель пристроя на базі кінцевого автомата з використанням

DOI 10.15588/1607-3274-2023-4-5

мови Verilog. Наукова новизна полягає у представлені способу вираження темпоральних подій на графі переходів кінцевого автомата, а також за допомогою HDL конструкцій під час розробки систем керування на CPLD та FPGA у система автоматизованого синтезу.

КЛЮЧОВІ СЛОВА: автоматний шаблон, мова опису апаратури, пристрої реального часу, темпоральні події, системи автоматизованого синтезу.

ЛІТЕРАТУРА

1. Baranov S. Logic and System Design of Digital Systems / S. Baranov. – Tallinn : TUT Press, 2008. – 267 p. doi.org/10.1016/B978-0-7506-8397-5.00005-2.
2. 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. doi.org/10.7551/mitpress/9657.001.0001 .
3. Design of real-time logic control system on FPGA / [M. Miroschnyk, A. Shkil, E. Kulak et al.] // Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS'19), September 13–16, Batumi, Georgia, 2019. – P. 488–491. doi.org/10.1109/ewdts.2019.8884387
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. doi.org/10.1023/a:1012392927006
5. Alur R. A theory of timed automata / R. Alur, D. L. Dill // Theoretical Computer Science. – 1994. – Vol. 126, No. 2. – P. 183–235. doi.org/10.1016/0304-3975(94)90010-8
6. 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. doi.org/10.1109/qsic.2011.30
7. Solov'ev V. V. Structural models of finite-state machines for their implementation on programmable logic devices and systems on chip / V. V. Solov'ev, A. S. Klimowicz // Journal of Computer and Systems Sciences International. – 2015. – Vol. 54, № 2. – P. 230–242. doi.org/10.1134/s1064230715010074
8. Design timed FSM with VHDL Moore pattern / [A. S. Shkil, M. A. Miroschnyk, E. N. Kulak et al.] // Radio Electronics, Computer Science, Control. – 2020. – №2(53). – P. 137–148. doi.org/10.15588/1607-3274-2020-2-14
9. Minimizing Deterministic Timed Finite State Machines / [D. Bresolin, A. Tvardovskii, N. Yevtushenko et al.] // In 14th IFAC Workshop on Discrete Event Systems WODES 2018. – IFAC-PapersOnLine, 2018. – Vol. 51, Issue 7. – P. 486–492. doi.org/10.1016/j.ifacol.2018.06.344
10. Equivalence checking and intersection of deterministic timed finite state machines / [D. Bresolin, K. El-Fakih, T. Villa, N. Yevtushenko] // Formal Methods in System Design – 2022. – № 7. – P. 1–26. doi.org/10.1007/s10703-022-00396-6
11. Tvardovskii A. S. Deriving homing sequences for finite state machines with timed guard / A. S. Tvardovskii, N. V. Yevtushenko // Automatic Control and Computer Sciences. – 2021. – Vol. 55, № 7. – P. 738–750. doi.org/10.3103/s0146411621070154
12. Ramparison M. TCTL Model Checking Lower/Upper-Bound Parametric Timed Automata Without Invariants / Ramparison Mathias, André Etienne, Lime Didier // Proc.International Conference Formal Modeling and Analysis of Timed Systems FORMATS 2018, September 4–6, Biejing, China, 2018. – P. 37–52. doi.org/10.1007/978-3-030-00151-3_3
13. Wagner G. An abstract state machine semantics for discrete event simulation / G. Wagner // Proc. 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. doi.org/10.1109/wsc.2017.8247830
14. Hardware implementation of timed logical control FSM / [A. Shkil, M. Miroschnyk, E. Kulak et al.] // Proc. of 2020 IEEE East-West Design & Test Symposium (EWDTS'20), Sept. 4–7, Varna, Bulgaria, 2020. – 6 p. [Електронний ресурс] / IEEE Xplore Digital Library – Режим доступу: www / URL: https://ieeexplore. ieee.org/document/ 9225129. doi.org/10.1109/ewdts50664.2020.9225129
15. Lamperti G. Introduction to Diagnosis of Active Systems / G. Lamperti, M. Zanella, Xiangfu Zhao. – Springer, 2018. – 353 p. doi.org/10.1007/978-3-319-92733-6
16. Ramparison M. TCTL Model Checking Lower/Upper-Bound Parametric Timed Automata Without Invariants / Ramparison Mathias, André Etienne, Lime Didier // Proc.International Conference Formal Modeling and Analysis of Timed Systems FORMATS 2018, September 4–6, Biejing, China, 2018. – P. 37–52. doi.org/10.1007/978-3-030-00151-3_3
17. Stéphane Lafortune Discrete Event Systems: Modeling, Observation, and Control / Stéphane Lafortune // Annual Review of Control, Robotics, and Autonomous Systems. 2019. – No. 2:1. – P. 141–159. doi.org/10.1146/annurev-control-053018-023659 .
18. Sabah Al-Fedaghi Modeling Physical/Digital Systems: Formal Event-B vs. Diagrammatic Thinging Machine / Sabah Al-Fedaghi // International Journal of Computer Science and Network Security. – 2020. – No. 20 (4). – P. 208–220. hal-02614504 , version 1 (20-05-2020).
19. Karl Wiegers. Software Requirements (Developer Best Practices) 3rd Edition / Karl Wiegers, Joy Beatty. – Developer Best Practices, 2013. – 672 p. doi.org/10.1109/9781118156629.ch3

DOI 10.15588/1607-3274-2023-4-5