Analysis for Design and Transformation of Autosynchronous State Machines

Michal KOVÁČ, Jaromír KOLOUCH

Dept. of Radio Electronics, Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic

xkovac03@stud.feec.vutbr.cz, kolouch@feec.vutbr.cz

Abstract. *The paper deals with design and transformation methodology of autosynchronous state machines. The result is the design methodology for autosynchronous state machines with one-hot and Gray encodings. On the basis of their simulation models the timing parameters are defined and conditions for the correct behavior are pointed out.*

In order to simplify the design of these state machines, the transformation methodology of synchronous state machine in VHDL at RTL level to autosynchronous state machine is designed. These transformed state machines are compared in their chip area, power consumption and timing.

Keywords

Autosynchronous circuit, state machine, VHDL, RTL, transformation, one-hot encoding, Gray encoding.

1. Introduction

Clock signal causes fairly big problems with increasing the circuit complexity and speed as well as with decreasing dimensions. Therefore the designers must look for new solutions from circuit level to system level. Current consumption, clock tree distribution and electromagnetic emissions are among the biggest issues. These problems are resolved by clock gating, by clock distribution net phasing and by partitioning to smaller system entities, like Globally Asynchronous Locally Synchronous (GALS) or Externally Asynchronous Internally Clocked (EAIC) systems independent on global clock reference. These techniques are not the most efficient solution of this problem. Therefore the designers go back over time to asynchronous systems, which eliminate synchronization problems.

The time representation is the main difference between synchronous and asynchronous systems. Asynchronous systems have no conception about common discrete time – synchronization. Asynchronous circuits use handshaking between their components to create required synchronization, communication and operations sequence. The main advantages of asynchronous systems are lower power consumption, higher computing speed, lower electromagnetic noise and better robustness to the variations of supply voltage, temperature and parameters in fabrication process. Asynchronous circuits have also disadvantages, hence attempts to create hybrid systems have been done. These hybrid systems can have advantages from both synchronous and asynchronous systems [6].

Autosynchronous circuits fall into the hybrid systems category. Their structure is similar to synchronous circuits, but the clock signal is generated locally on the basis of information about toggling the edge triggered flip-flop. Because these circuits are based partially on synchronous circuit principle their design is simpler than asynchronous circuits design. Likewise transformation from synchronous circuits to autosynchronous circuits is simple. Although there exist several principles of autosynchronous circuits (e.g. EAIC), their design is complex and parameters are not often competitive in comparison with other circuits. Therefore this article looks for a new principle of autosynchronous state machines design.

2. Autosynchronous State Machines: An Overview

Autosynchronous sequential circuits belong to a small group under the asynchronous sequential circuits. Hence the main feature is ability to self-control without any global clock authority. These circuits look like synchronous circuits from inside aspect because they include clocked state register. The principle is illustrated in Fig. 1.

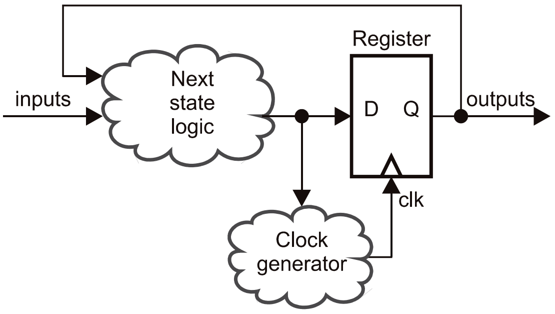

Fig. 1. Autosynchronous state machine principle.

The circuit works like a standard synchronous state machine, but it generates the clock signal itself on the basis of information about settlement of the next state. Additional logic is necessary to generate the clock. This logic determines when the current state is stable and simultaneously generates clock pulse for the state register.

3. Analysis

The state code assignment is the most important point for autosynchronous circuit design which differs from synchronous circuit design process. The rest of points of design process stay the same as in synchronous circuit design process. Moreover, the decision logic must be added. This logic detects the moment for toggling the state register – the stable state, described in the following sections.

Source [1] refers to three ways of state assignments. The first way is encoding with minimal bit change, which meet Gray and Johnson code. The second way is priority neighborhood of the states in Karnaugh map assignment. The third way is one-hot encoding, where one state represents one flip-flop. Simultaneously the assignment method should be chosen, which avoids the hazards and critical races [2].

The Gray encoding and one-hot encoding have appeared as the most suitable encodings on the basis of previous knowledge. These encodings were analyzed on the simple state machine example. Fig. 2 shows the frame of simple six-state machine with one-hot encoding. This state machine was created only as an illustrative example without an intended application purpose. As we can show in Fig. 2, there are two possible intermediate states for the transition between two adjacent states in one-hot encoding state machine. Signals don't change ideally simultaneously in a real environment. There is always a minimal time gap between signals. Therefore the transition between two adjacent states leads through an intermediate state. Fig. 2 demonstrates that this intermediate state is known and has exactly two possibilities - either all zeros or two ones.

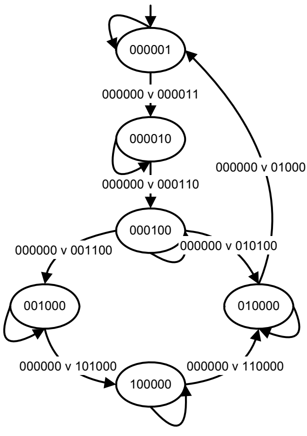

Fig. 2. Intermediate states in one-hot encoding scheme.

From this knowledge we can decide when the operating state occurs by the modulo-2 mutual addition of single bits of the next state vector. This operation is carried out by XOR function:

$$y = D(0) \oplus D(1) \oplus .. \oplus D(n-1) \tag{1}$$

where D is the next state vector and n is the state vector bit width. This function equals to zero when intermediate state occurs, otherwise equals to one. Source [3] eliminates the uncertainty of two possible intermediate states by an additional term in the next state equation (the second term in the equation):

$$Y_j = \sum_{k=0}^{m-1} y_k . f_{j \leftarrow k} + y_j . F_j \tag{2}$$

where F_j is the Boolean sum of all active y-variables in states to which the jth state transits, m is a number of next state functions.

The Gray encoding has even better properties, where only one bit changes between two neighboring states and no glitches occurs on the combinational logic output. Further advantage is that Gray encoding can cover ideally all possible states in contrast to one-hot encoding. But state diagrams with complex transitions make impossible to change only one state variable between neighboring states. Such assignment is then unsuitable in respect of critical races and glitches generation.

Even though hazards occur in next state combinational logic at the one-hot encoding their identification is simple and the stable state moment is also well identifiable. Combinational logic design for one-hot encoding is simple. On the other side, Gray encoding has more profitable properties but in complex cases the only one bit change between neighboring states can't be achieved. Hence the one-hot encoding is more suitable for practical applications.

4. Design Methodology

The methodology for autosynchronous circuits design was created. It is based on well known methodology for synchronous circuit design [1], [3]. On the basis of the previous analysis the new steps in design methodology of autosynchronous state machines were created and added for certain state encodings. These steps contain additional logic for determination the toggle moment. This logic is called stable state detector.

As it was shown in previous analysis stable state detection is primarily suitable for Gray and one-hot encoding. The next state vector change detection is the principle of stable state finding based on comparison of the next state vector with the current state vector. The state machine with all stable states is assumed thence the next state vector change is stimulated only by input signals change.

For stable state detection at both encodings (one-hot, Gray) the simple comparison of the current state with the next state is realized by using logical sum of single bits modulo-2 addition (XOR) of the current state and next state vectors:

$$y = \bigvee_{i=0}^{n-1} D(i) \oplus Q(i) \tag{3}$$

where Q is the current state vector and i is bit index.

One-hot encoding needs moreover to control the transition moment through the intermediate state given by (1). Logical product of these two functions (1) and (3) determines the appropriate moment for triggering clock pulse.

Many other methods exist for detection the toggling moment of the state register. Except the basic known discrete methods like the global clock in synchronous systems and matched delay in asynchronous systems, it's possible to use NCL logic [4]. Another EAIC systems [5] use special flip-flops with synchronizing output signal.

The design methodology was verified on the simple state machine samples with Gray and one-hot encoding. These two encodings are not the only ones which may be used for autosynchronous state machine design. But they are the most appropriate for optimized additional logic design. The additional logic would be more complex for the other encodings. Autosynchronous state machines design methodology assumed fundamental mode operations for simplicity. Securing the fundamental mode is an extensive issue for purpose of this article.

5. Timing Properties

The autosynchronous state machine timing model, depicted in Fig. 3, was created on the basis of previous simulations.

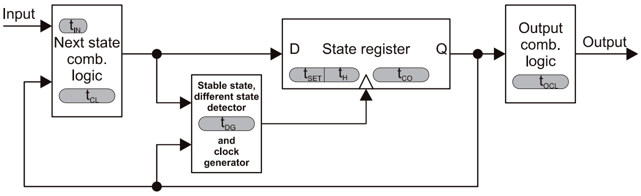

Fig. 3. Time model of autosynchronous state machine.

Time parameters in Fig. 3 are: t_{IN} – time for inputs stabilization, t_{CL} - next state combinational logic delay, t_{DG} – stable state detector combinational logic delay, t_{SET} – flip-flop setup time, t_H – flip-flop hold time, t_{CO} – flip-flop clock to output delay, t_{OCL} - output combinational logic delay.

Contrary to the synchronous state machine behavior autosynchronous state machine responds to inputs or states changes. The timing diagram in Fig. 4 shows the autosynchronous state machine behavior after the inputs change. After the input signal stabilization represented by time interval t_{IN} the next state combinational logic generates a new next state after the delay t_{CL}. The current state was the same as the next state till now. The newly generated next state now differs from the current state. This change is detected by the stable state detector and it generates a clock pulse for the state register after combinational logic delay t_{DG}.

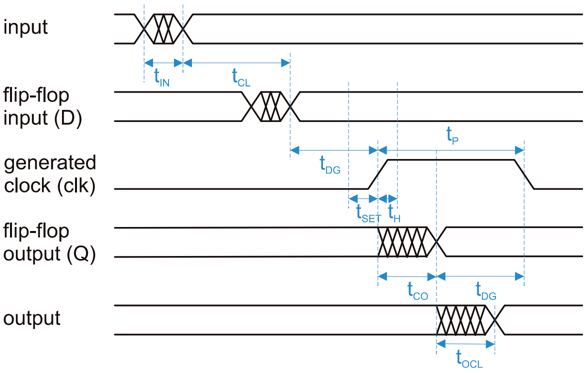

Fig. 4. Autosynchronous state machine timing diagram.

The width of generated clock pulse depends only on logic delays:

$$t_P = t_{CO} + t_{DG}\,. \tag{4}$$

The throughput delay from input to output is well defined from the timing diagram as:

$$t_{PROP} = t_{CL} + t_{DG} + t_{CO} + t_{OCL}\,. \tag{5}$$

To determine the performance, it is necessary to analyze the timing diagram for two states transition. Fig. 5 shows this timing between two stable states. The performance is defined as the interval between two input stimulus from external environment. Further the minimal time gap t_{CI} between two input changes is defined for circuit performance evaluating. Some assumptions and moments in time diagram are necessary to establish using equations. Fundamental mode will be assumed for simplification, i.e. only one input can change at a time under the stable conditions (t_{IN} parameter will be zero) and the first input change will be at time $t_1 = 0$. For minimal time gap between two inputs changes it is necessary to determine certain conditions. The first condition is that the next input change (d_2) occurs on the next state combinational logic output (flip-flop input D) after the hold time t_H of the past clock pulse (cv_1). The second condition is that the rising edge of the following clock pulse (c_2) doesn't overtake the falling edge of the past clock pulse (f_1). Under mentioned moments will be defined for these conditions depicted also in Fig. 5.

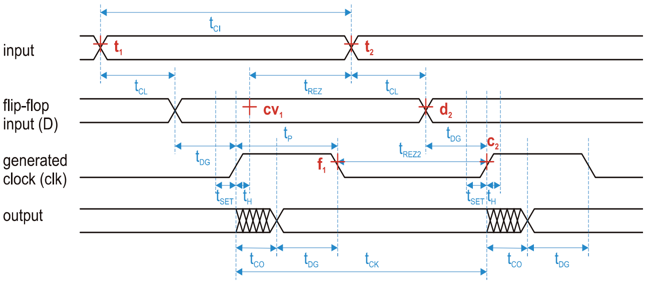

Fig. 5. Timing diagram of transition between two stable states.

The second stabilization of next state combinational logic output:

$$d_2 = t_2 + t_{CL} \tag{6}$$

where t_2 is the time of the second inputs change.

The moment of the new data validity on the flip-flop input after the first rising edge:

$$cv_1 = t_{CL} + t_{DG} + t_H \,. \tag{7}$$

The rising edge moment of the second clock pulse:

$$c_2 = t_2 + t_{CL} + t_{DG} \,. \tag{8}$$

The falling edge moment of the first clock pulse:

$$f_1 = t_{CL} + 2.t_{DG} + t_{CO} \,. \tag{9}$$

On the first condition and determined moments basis, the equations for minimal range of the inputs change derivation are expressed:

$$\begin{aligned} & d_2 > cv_1 + t_{REZ} \Rightarrow \\ & t_2 + t_{CL} > t_{CL} + t_{DG} + t_H + t_{REZ} \Rightarrow \\ & t_2 > t_{DG} + t_H + t_{REZ}, \end{aligned} \tag{10}$$

where t_{REZ} is a time reserve for these conditions achievement.

The second condition can be described as:

$$\begin{aligned} & c_2 > f_1 + t_{REZ2} \Rightarrow \\ & t_2 + t_{CL} + t_{DG} > t_{CL} + 2.t_{DG} + t_{CO} + t_{REZ2} \Rightarrow \\ & t_2 > t_{DG} + t_{CO} + t_{REZ2}, \end{aligned} \tag{11}$$

where t_{REZ2} is a safety time interval between the first clock pulse falling edge and the second clock pulse rising edge. For the reason that t_{CO} time in (11) is longer than t_H time in (10) (for all flip-flops in whatever technology) the second equation (11) will be chosen from these two conditions. The minimal inputs change period is derived from the previous chosen equation (11):

$$\begin{aligned} & t_{CI} > t_{2\min} - t_1 \Rightarrow \\ & t_{CI} > t_{DG} + t_{CO}, \end{aligned} \tag{12}$$

which is simultaneously the minimal clock period t_{CK} for comparison with synchronous circuits. It must be taken into account that equality at both equation sides makes the past falling clock edge with the following rising clock edge overlapping. Hence it is necessary to use time reserve t_{REZ2} which provide for the stable function of the state machine:

$$t_{CI} = t_{DG} + t_{CO} + t_{REZ2} \,. \tag{13}$$

For 50 % duty cycle the time reserve should be:

$$t_{REZ2} = t_{DG} + t_{CO} \,. \tag{14}$$

That means the cycle period is double of that (12):

$$t_{CK} = t_{CI} = 2.(t_{DG} + t_{CO}) \,. \tag{15}$$

6. Transformation Methodology

Design of systems with other than global clock control synchronization principle is complicated. Therefore the advantageous manner is transformation of a known synchronous system to a desired target system with different synchronization authority with similar function. The most efficient way is transform at RTL level. This Register Transfer Level is technology independent description on the high-level abstraction. The advantage is that optimization like the logical functions minimization and gate assigning for certain technology occurs after this process step in synthesis [7].

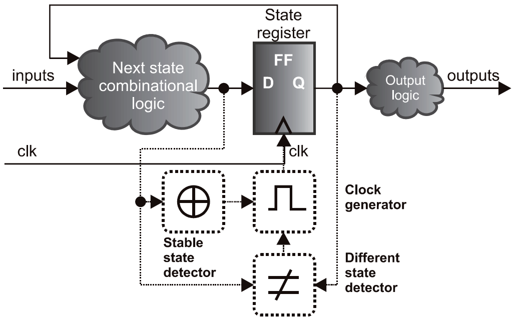

Fig. 6. State machine block diagram with added blocks after transformation to autosynchronous state machine.

Synchronous state machine description in VHDL on RTL level abstraction is assumed.

A simple transformation was created which actually adds combinational logic blocks for local clock signal generation as it can be seen in Fig. 6. These blocks are dashed. Remaining blocks represent original synchronous scheme. Stable state detector block is combinational logic for stable state detection. The different state detector block is combinational logic which compares the current state with the next state vector. The clock generator block evaluates outputs of both detectors and generates the clock pulse.

The designed transformation methodology is as follows:

1. Numbering of states detection in “type” definition of VHDL code. Substituting these states by constants in one-hot encoding.
2. Conversion of the state vectors to a defined width bit vectors according to number of states and encoding type (state, next_state).
3. Adding the different state detector process and declarations of related signals.
4. Adding the stable state detector process and declarations of related signals (for one-hot encoding only).
5. Adding the clock generator as the combinational equation to process outputs from blocks described in item 4 and 5.
6. Redeclaration of clock from port to internal signal.

This methodology could be also automated as a script in future.

This section dealt with autosynchronous state machine transformation from synchronous version on RTL level in VHDL language. The advantage of designed transformation is simplicity of conversion by adding a part of combinational logic.

7. Comparison

The transformation of 6-state (sa6_synchr_1zN) and 15-state (sa15_synchr_1zN) synchronous Moore machines with one-hot encoding to autosynchronous state machines (sa6_asynchr_1zN) respectively (sa15_asynchr_1zN) was realized on the basis of the introduced transformation methodology. Alternatives of synchronous state machines with Gray and binary state encoding were created for additional comparison (sa6_synchr_bin, sa6_synchr_gray), (sa15_synchr_bin, sa15_synchr_gray) respectively.

Three basic criteria were used in order to compare the properties of state machines which are the main factor in digital system choice. These criteria are: chip area, power consumption and time performance. Properties of designs were compared in ISE 10.1 Xilinx development tool over the technology libraries for Spartan3 XC3S200-5pq208 gate array.

Due to FPGA target platform the chip area can be compared by numbers of LUT tables, flip-flops FF and slices. Chip area comparison results are depicted in Fig. 7 by number of LUTs.

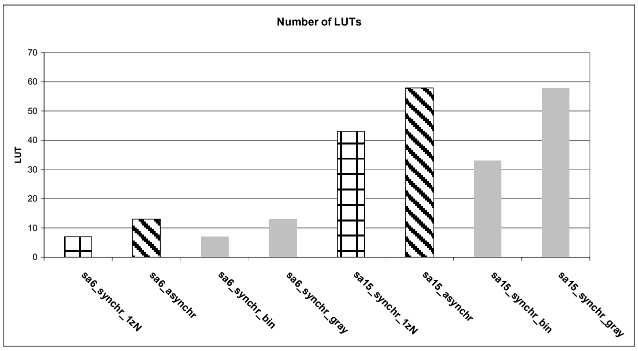

Fig. 7. Number of LUT comparison.

These values were obtained from XST synthesis tool. Checked columns represent synchronous machines and banded columns represent autosynchronous machines. The comparison is enriched by gray columns with another state encodings (Gray, binary).

The power consumption was analyzed in XPower tool at design process step after mapping. The FPGA parameters were: $V_{CCINT} = 1.2$ V, $V_{CCAUX} = 2.5$ V, $V_{CCO25} = 2.5$V and 100 MHz clock frequency. This analysis was corrected by VCD files generated from testbenches. In Fig. 8 and 9, the logic power and total dynamic power are compared.

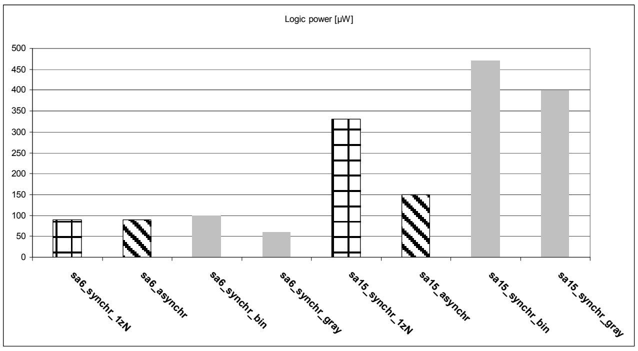

Fig. 8. Logic power comparison.

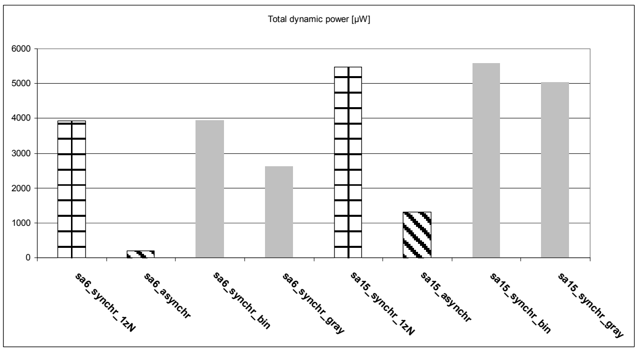

Fig. 9. Total dynamic power comparison.

Static time analysis was realized in Timing Analyzer tool in order to analyze the state machines time delays at the implementation process step after mapping. Fig. 10 shows the minimal period cycle comparison for different state machine types. These values were taken from parameter Clock to setup on destination clock x (signal x is a state machine input) in Timing Analyzer.

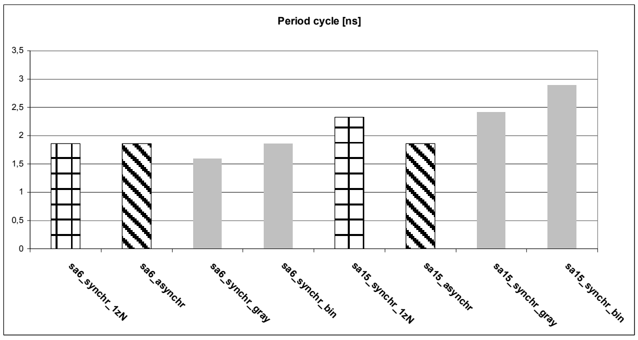

Fig. 10. Cycle period comparison.

The transformed state machines were compared with their synchronous originals in parameters like chip area, power consumption and timing in order to verify transformation and design methodology.

The required chip area of the autosynchronous state machines is larger than that for the synchronous state machines due to additional combinational logic. Compared to synchronous state machines the autosynchronous state machines requirement for LUT was about by 85 % larger in 6-state machine and about by 34 % larger in 15-state machine respectively.

In power consumption case the autosynchronous machines have better properties due to global clock inactiveness. The 6-state and 15-state autosynchronous machines had the logic power consumption 9-times less and twice less respectively.

As to timing parameters of autosynchronous state machines, they were similar to these of synchronous state machines or better for complex designs. The minimal period cycle was similar at 6-state machines, but in 15-state autosynchronous machines case the minimal cycle period was by 25 % less that of the synchronous state machine. The throughput delays from input to output were larger for autosynchronous state machines by 34 % in 6-state machine and by 43 % in 15-state machine.

8. Conclusion

In this paper the design and transformation methodology of autosynchronous state machines were presented. On the basis of state assignments analysis the design methodology of autosynchronous state machines with one-hot and Gray encodings was created. The timing parameters of the designed state machine were determined. As the second way for the autosynchronous state machines design the transformation in VHDL at RTL level abstraction was designed. The transformed state machines were compared in chip area, power consumption and timing. From the system perspective autosynchronous circuits are innovative alternative to synchronous and asynchronous approaches. These systems take advantages from both approaches. But they have also disadvantages. Interfacing autosynchronous controllers in large systems is not a simple task. Therefore the synchronous systems with modified clock tree are mostly used. It has comparable parameters with autosynchronous systems. Hence the autosynchronous systems aren't widely used in practical applications. This work was an introductory study for comparison of systems with different synchronizations in order to reduce power consumption and eliminate problems with clock distribution. This article brings practical information about the state machine design with different synchronization principle.

Acknowledgement

The work described in this paper was supported by the research program of the Ministry of Education of the Czech Republic No. MSM 0021630513 "Advanced Electronic Communication Systems and Technologies" (ELCOM), and by GA CR (Czech Science Foundation) projects No. 102/08/H027 and No.102/08/0851.

References

[1] REICHARDT, J., SCHWARZ, B. *Digital Systems: Handout*. 3rd edition. Hamburg: University of Applied Sciences. 2004. 150 p.

[2] FUHRER, M. R., LIN, B., NOWICK, M. S. Algorithms for the optimal state assignment of asynchronous state machines. In *16th Conference on Advanced Research in VLSI*. Chapel Hill (North Carolina, USA), 2001, p. 59-75. ISBN-13: 9780818670473.

[3] TINDER, F. R. *Engineering Digital Design*. 2nd edition, revised. Academic Press-Elsevier, 2000. 884 p. ISBN-13: 978-0-12-691295-1.

[4] FANT, K., BRANDT, S. Null convention logic, a complete and consistent logic for asynchronous digital circuit synthesis. In *International Conference on Application Specific Systems, Architectures, and Processors (ASAP '96)*. Chicago (Illinois, USA), 1996, p. 261–273. ISBN 0-8186-7542-X.

[5] VANSCHEIK, S. W., TINDER, F. R. High speed externally asynchronous/internally clocked systems. *IEEE Transactions on Computers*, 1997, vol. 46, no. 7, p. 824-829, ISSN 0018-9340.

[6] SPARSO, J., FURBER, S. *Principles of Asynchronous Circuit Design - A System Perspective*. Boston: Kluwer Academic Publishers, 2001. ISBN 0-7923-7613-7.

[7] CORTADELLA, J., KONDRATYEV, A., LAVAGNO, L., SOTIRIOU, P. CH. Desynchronization: synthesis of asynchronous circuits from synchronous specifications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2006, vol. 25, no. 10, p. 1904-1921, ISSN 0278-0070.

About Authors...

Michal KOVÁČ (*1982 Piešťany, Slovakia) graduated at the Faculty of Electrical Engineering and Communication (FEEC), Brno University of Technology (BUT), in 2006. Since then he has been a Ph.D. student at the Dept. of Radio Electronics, FEEC, BUT.

Jaromír KOLOUCH was born in Brno, Czech Republic, in 1945. Currently, he is an Associated Professor with the Department of Radio Electronics, Brno University of Technology, Czech Republic. His scientific interest is focused on pulse and digital technology, digital design, programmable logic.