

The Application and Adaptation of the Two Sources of Code and Natural Encoding Method for Designing a Model of Microprogram Control Unit with Base Structure

Łukasz Smoliński*, Alexander Barkalov, Larysa Titarenko

Institute of Computer Engineering and Electronics, University of Zielona Góra, Zielona Góra, Poland
Email: *L.Smolinski@weit.uz.zgora.pl

Received 27 September 2014; revised 28 October 2014; accepted 6 November 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The article presents a modification to the method which applies two sources of data. The modification is depicted on the example of a compositional microprogram control unit (CMCU) model with base structure implemented in the complex programmable logic devices (CPLD). First, the conditions needed to apply the method are presented, followed by the results of its implementation in real hardware.

Keywords

CPLD, PAL, UFM, CLB, Two Sources of Code, One-Hot Encoding, Natural Encoding, CMCU

1. Introduction

Control units are very important parts of digital systems [1]-[6]. Nowadays, complex programmable logic devices (CPLD) are widely used for implementing logic circuits of control units [7] [8]. However, the issue of reducing the size of a control unit is still a subject of current interest. Finding a solution to this issue will make it possible to improve such indicators as the speed of performance and the power consumption of the designed system [9] [10]. CPLDs include macrocells of programmable array logic (PAL) with a limited number of terms.

*Corresponding author.

To reduce the amount of hardware in the logic circuit of a control unit, the peculiarities of the CPLD and the features of a control algorithm to be implemented should be taken into account. If a control algorithm is represented by a linear graph-scheme of the algorithm (GSA), then a model of the compositional microprogram control unit (CMCU) can be used for its interpretation [11]. It is assumed by the authors that the control memory (CM) of the CMCU can be implemented as external PROM/ROM memory. Some of the CPLD family devices are equipped with integrated memory. For example, Altera CPLD devices are equipped with user flash memory (UFM) [12], whereas Cypress CPLD devices are equipped with cluster memory blocks (CMB) [13]. This article presents a mathematical model and a design algorithm with two sources of code and natural encoding which has been modified and adopted to CMCU model with base structure [14] [15].

2. Background of the CMCU with Base Structure and Two Sources of Code

It is assumed that the GSA is represented by sets of vertices B where $B = B_1 \cup B_2 \cup \{b_0, b_E\}$ and a set of arcs E where $E = \{\langle b_i, b_q \rangle\}$. It is further assumed that $b_i, b_q \in B$ and b_0 is an initial vertex, b_E is a final vertex, B_1 is a set of operator vertices, B_2 is a set of conditional vertices. A vertex contains a microinstruction $m_i \in M$ ($i = 1, \dots, |B_1|$) and m_i is a set of data-path microoperations $Y = \{y_1, \dots, y_N\}$ [16]-[18]. Each vertex $b_i \in B_2$ contains a single element x_i of a set of logical conditions $X = \{x_1, \dots, x_L\}$. A set C of operational linear chains (OLC) for the GSA shall be formed, where each OLC is a sequence of operator vertices and each pair of its adjacent components corresponds to an arc of the GSA.

$$C = \{\alpha_1, \dots, \alpha_i\}. \quad (1)$$

Each OLC α_i has only one output O_i and an arbitrary number of inputs I_i^j . The α_i elements are understood as a state of the system which is identified by O_i elements. OLC outputs make up the collection $O_i \in \Pi_C$. Formal definitions of OLC, its input and output can be found in [11]. Each vertex from $b_i \in B_1$ corresponds to microinstruction $m_i \in M$ stored in the control memory (CM) of CMCU and it has an address $A(b_i)$. The microinstructions can be addressed using R bits, where

$$R = \lceil \log_2 |M| \rceil, \quad |M| = |B_1| \quad (2)$$

and the bits are represented by variables from the set T :

$$T = \{T_1, \dots, T_R\}. \quad (3)$$

Assuming that the $\alpha_g \in C$ OLC includes F_g components, the following condition takes place:

$$A(b_{g_{i+1}}) = A(b_{g_i}) + 1 \quad (i = 1, \dots, F_g - 1). \quad (4)$$

In Equation (4) b_{g_i} is the i -th component of OLC. The first source of information about the state of the system is the register RG. The Value of used flip-flops needed for the implementation RG is R_1 , where:

$$R_1 = \lceil \log_2 M_1 \rceil, \quad M_1 = |C|. \quad (5)$$

The flip-flops outputs are represented by variables from the set τ :

$$\tau = \{\tau_1, \dots, \tau_{R_1}\}. \quad (6)$$

The second source of information about the state of the system is the unused space of the control memory (CM) [19] [20]. An external PROM chip or memory integrated with the CPLD may be applied to implement the CM. The memory has t outputs, where $t = 2, 4, 8, 16$ [21]. Some information can be implemented using free outputs of the CM. it is assumed that one-hot encoding of microoperations is used [22]-[26]. The CM word has R_0 bits, where

$$R_0 = N + 2. \quad (7)$$

Thus, the number of available unused flip-flops is:

$$R_2 = t - R_0. \quad (8)$$

The R_2 bits are represented by variables from the set P :

$$P = \{p_1, \dots, p_{R_2}\}. \quad (9)$$

If the condition

$$R_2 > 0 \quad (10)$$

takes place, the method can be used. The number of repetitions of each state occurring as the initial state is counted in the transitions of the algorithm. The states of the system are identified through the outputs O_i of the operational linear chains. The outputs belong to the collection Π_C . The R_2 most used O_i items shall be taken. Next, these items shall be moved to a new collection $O_i \in \Pi_B$. Following the above assumptions, an equation

$$R_2 \geq |\Pi_B| \quad (11)$$

takes place. The collection Π_C has been divided into two collections Π_A and Π_B , where $\Pi_C = \Pi_A \cup \Pi_B$. The control memory CM is a source of the variables $p_i \in P$ applied to encode $O_i \in \Pi_B$ by the code $K_B(O_i)$. The register RG is a source of the variables $\tau = \{\tau_1, \dots, \tau_{R_1}\}$ applied to encode $O_i \in \Pi_A$ by the code $K_A(O_i)$. Provided that the assumption presented in equation (11) is fulfilled, it is possible to apply the one-hot encoding to the elements from the collection $O_i \in \Pi_B$.

The combinational circuit (CC) generates data Φ for the counter (CT) and data Ψ for the register (RG):

$$\Psi = \Psi(P, X, \tau), \quad (12)$$

$$\Phi = \Phi(P, X, \tau). \quad (13)$$

The synthesis of the CMCU algorithm includes the following steps:

- 1) Forming the set of OLCs.
- 2) Forming the control memory content.
- 3) Forming the transition table of the CMCU.
- 4) Counting the occurrence of $O_i \in \Pi_C$ on the basis of the transition.
- 5) Dividing the collection Π_C into two collections Π_A and Π_B .
- 6) Updating the control memory content.
- 7) Forming the excitation functions for the CT and RG with using two sources of code.
- 8) Synthesis of the logic circuit of CMCU.

Figure 1 presents a structural diagram of the CMCU with two sources of code. The pulse Start causes loading of the first microinstruction address into a counter CT and set up of a fetch flip-flop TF. If Fetch = 1, then microinstructions can be read out from the control memory CM. If a current microinstruction does not correspond to an OLC output, then a special variable y_0 is formed together with microoperations $Y_q \subseteq Y$. If $y_0 = 0$, then content of the CT is incremented according to the addressing mode. Otherwise, the block of combinational circuit logic (CC) generates functions Φ and Ψ . If $y_E = 1$, then the CMCU stops and new data from CM is not loaded.

3. Main Idea behind the Proposed Method

The excitation functions generating data for the counter CT and register RG in Equation (12) and Equation (13)

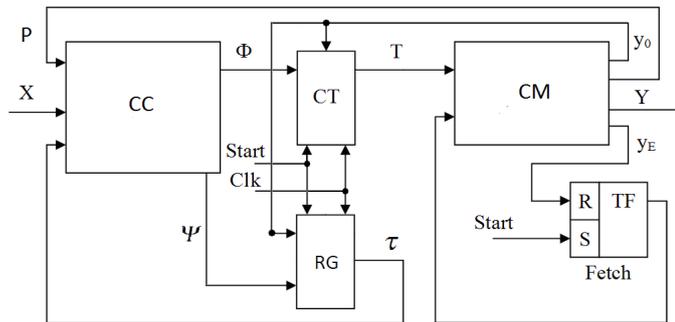


Figure 1. Structural diagram of CMCU U_1 .

are built with the use of logical terms. Let F_Ψ and F_Φ be such collections of excitation functions required to realize Equation (12) and Equation (13), that:

$$F_\Psi = \{\tau'_1(P, X, \tau), \dots, \tau'_{R_1}(P, X, \tau)\}, \quad (14)$$

$$F_\Phi = \{T_1(P, X, \tau), \dots, T_R(P, X, \tau)\}. \quad (15)$$

Functions from the collections F_Ψ and F_Φ generate data represented by the variables depicted in Equation (3) and Equation (4). The terms which are used for building the excitation functions are created by joining two parts: the conjunction of conditions $X = \{x_1, \dots, x_L\}$ and the code $K_B(O_i)$ or $K_A(O_i)$. The length of the terms has a direct influence on the number of logical gates used in implementing the CMCU model, and the number of the gates used has, in turn, a direct influence on the size of the realized system. For the base method the maximum length of a term built with the use of the information from the collection $O_i \in \Pi_A$ equals:

$$L_A = |X| + R_1. \quad (16)$$

The maximum length of a term built with the use of information from the collection $O_i \in \Pi_B$ equals:

$$L_B = |X| + 1. \quad (17)$$

The second factor which has a direct influence on the size of the designed system is the possibility of minimizing the excitation functions for the counter CT and register RG. As far as the method with two sources of code is concerned, this possibility is degraded under negative conditions, compared to the method with one source of code. For example, having the OLC elements encoded as in **Table 1** and the function for the method with one source of code designed in the following way:

$$\begin{aligned} \tau'_{1 \text{ one-source}} &= K_C(O_{33}) * x_1 + K_C(O_2) * x_1 + K_C(O_3) * x_1 + K_C(O_4) * x_1 + K_C(O_{11}) * x_1 + K_C(O_{12}) * x_1 \\ &= \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 \\ &\quad + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 \\ &= \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1. \end{aligned}$$

The excitation function $\tau'_{1 \text{ one-source}}$ requires 17 logic gates to be implemented. Let us assume that the system CM has $R_2 = 3$ of unused outputs. In the case of the method with two sources of code and one-hot encoding, as in **Table 2** the function $\tau'_{1 \text{ one-source}}$ shall take the form of $\tau'_{1 \text{ one-hot}}$ to realize the same task:

$$\begin{aligned} \tau'_{1 \text{ one-hot}} &= K_A(O_{33}) + K_A(O_2) * x_1 + K_B(O_3) * x_1 + K_A(O_4) * x_1 + K_B(O_{11}) * x_1 + K_B(O_{12}) * x_1 \\ &= \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + p_1 * x_1 \\ &\quad + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + p_2 * x_1 + p_3 * x_1 \\ &= \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + p_1 * x_1 + p_2 * x_1 + p_3 * x_1. \end{aligned}$$

For the analyzed example, the excitation function $\tau'_{1 \text{ one-hot}}$ requires 18 logic gates to be implemented. Provided that the length of the terms has been reduced according to Equation (17), the possibility of minimizing the excitation functions has been degraded. This results from the peculiarity of one-hot encoding. With such encoding, the terms which use elements from the collection Π_B do not undergo minimization. Additionally, minimizing two terms which are built on the basis of information from various sources is also impossible, which leads to an increase in the level of degradation of the possibility of minimizing excitation functions. In order to reduce the level of degradation of the possibility of minimization of excitation functions, the mathematical model and design algorithm with natural encoding has been adopted to the CMCU with base structure [14] [15]. First, natural encoding will be used for encoding information about the state from the CM. In such a case we can use I_{CB} information from CM, where

$$I_{CM} = 2^{R_2} - 1. \quad (18)$$

In Equation (18), 1 is subtracted to take into account the case when microinstruction is not the source of code. This will be marked by code zero. This code will be not used in excitation functions. To encoding $O_i \in \Pi_B$

Table 1. OLC items with base structure.

OLC outputs	$K_C(O_i)$						
	O_i	τ_6	τ_5	τ_4	τ_3	τ_2	τ_1
O_1	0	0	0	0	0	0	0
O_2	0	0	0	0	0	0	1
O_3	0	0	0	0	0	1	0
O_4	0	0	0	0	0	1	1
...							
O_{11}	0	0	1	0	1	1	0
O_{12}	0	0	1	0	1	1	1
...							
O_{33}	1	0	0	0	0	0	0

Table 2. OLC items with two sources of code and one-hot encoding.

OLC outputs	$K_A(O_i), K_B(O_i)$									
	O_i	τ_6	τ_5	τ_4	τ_3	τ_2	τ_1	p_3	p_2	p_1
O_1	0	0	0	0	0	0	0	-	-	-
O_2	0	0	0	0	0	0	1	-	-	-
O_3	-	-	-	-	-	-	-	-	-	1
O_4	0	0	0	0	0	1	1	-	-	-
...										
O_{11}	-	-	-	-	-	-	-	-	1	-
O_{12}	-	-	-	-	-	-	-	1	-	-
...										
O_{33}	1	0	0	0	0	0	0	-	-	-

elements is used natural encoding. In order to reduce the maximum length of the terms built on the basis of the collection $O_i \in \Pi_B$ the following limitation is introduced:

$$R_2 < R_1. \tag{19}$$

In such a case, the maximum length of a term built with the use of information from the collection $O_i \in \Pi_B$ equals:

$$L'_B = |X| + R_2. \tag{20}$$

In the analyzed example for the method with two sources of code and natural encoding, as in **Table 3**, the function $\tau'_{1 \text{ one-source}}$ shall become $\tau'_{1 \text{ natural}}$ in order to realize the same task:

$$\begin{aligned} \tau'_{1 \text{ natural}} &= K_A(O_{33}) * x_1 + K_B(O_2) * x_1 + K_B(O_3) * x_1 + K_B(O_4) * x_1 + K_B(O_{11}) * x_1 + K_B(O_{12}) * x_1 \\ &= \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + p_3 * p_2 * p_1 * x_1 + p_3 * p_2 * p_1 * x_1 + p_3 * p_2 * p_1 * x_1 \\ &\quad + p_3 * p_2 * p_1 * x_1 + p_3 * p_2 * p_1 * x_1 \\ &= \tau_6 * \tau_5 * \tau_4 * \tau_3 * \tau_2 * \tau_1 * x_1 + p_3 * p_1 * x_1 + p_2 * x_1. \end{aligned}$$

The excitation function $\tau'_{1 \text{ natural}}$ requires 11 logic gates to be implemented. Applying natural encoding has made it possible to reduce the level of degradation of the possibility of minimizing of excitation functions. Natural encoding applied to the elements from collection Π_B makes it possible to minimize the terms built on the

Table 3. OLC items two sources of code and natural encoding.

OLC outputs	$K_A(O_i), K_B(O_i)$									
	O_i	τ_6	τ_5	τ_4	τ_3	τ_2	τ_1	p_3	p_2	p_1
O_1	0	0	0	0	0	0	0	-	-	-
O_2	-	-	-	-	-	-	-	0	0	1
O_3	-	-	-	-	-	-	-	0	1	0
O_4	-	-	-	-	-	-	-	0	1	1
...										
O_{11}	-	-	-	-	-	-	-	1	1	0
O_{12}	-	-	-	-	-	-	-	1	1	1
...										
O_{33}	1	0	0	0	0	0	0	-	-	-

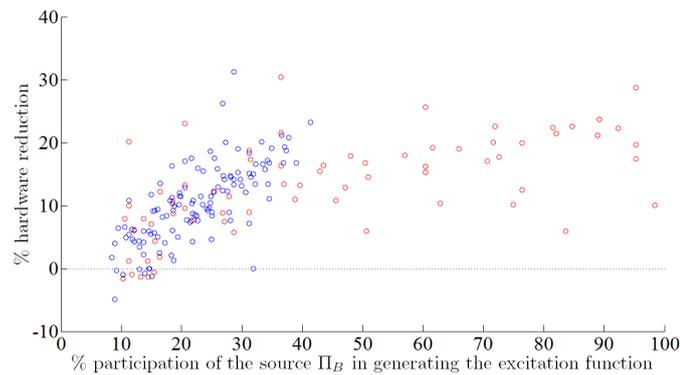


Figure 2. Reduction in the size of the system due to the participation of the source.

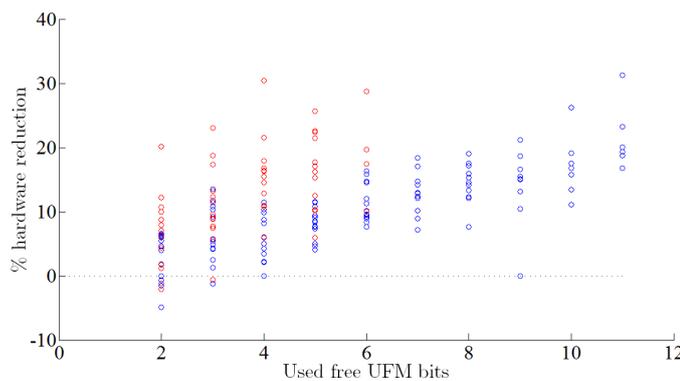


Figure 3. Reduction in the size of the system by free bits of memory.

basis of data from this collection. Furthermore, using natural encoding enables transferring a greater amount of information to the collection (source) Π_B . Fulfilling the criteria of Equation (19) ensures reduction in the maximum length of the terms, in accordance with Equation (20).

4. Results and Conclusions

Figure 2 and Figure 3 present the results of the implementation of the model in real hardware. The Altera family MAX II device EPM1270 F256C5 equipped with UFM has been used for tests. The results presented in

Figure 2 and **Figure 3** have been obtained from research conducted on 200 randomly generated CMCU models with the following properties: $45 \leq |\Pi_C| \leq 84$, $5 \leq R \leq 7$, $5 \leq R_0 \leq 16$, $6 \leq R_1 \leq 7$, $0 \leq R_2 \leq 11$, $t = 16$, $16 \leq X \leq 28$, $3 \leq Y \leq 13$, $115 \leq \text{Transitions} \leq 196$. Red points presented in **Figure 2** and **Figure 3** show the results obtained with the use of the natural method, whereas the blue points refer to the results obtained with the one-hot method. **Figure 2** depicts the relationship between the obtained reduction in the size of the system (in percentage) and the participation of the source of data Π_B in generating the excitation functions (in percentage). The application of natural encoding made it possible to avoid the effect of degradation of the possibility of minimizing the functions. A linear interrelation between the participation of the source Π_B and the obtained reduction in the size of the system has been observed. In the analyzed models with a small participation of the source Π_B —within the scope of up to 17%, deterioration of the possibility of minimizing the functions has been observed, which results in an increase in the size required for the realization of the designed system.

Figure 3 presents the relationship between the obtained reduction in the size of the system (in percentage) and the value of free bits used to representation information from the source Π_B . As can be seen in **Figure 3**, in the case of both the methods there occurs degradation of the possibility of minimizing the functions under negative conditions. The results of conducted experiments indicate that the ability to minimize logic terms and the ability to reduce the length of the terms, which follows from Equation (19) and is characteristic of the natural method, makes it possible to obtain better effects of reducing the system than the ability to reduce the length of logic terms following from Equation (17), yet without the possibility of minimizing the terms from the source Π_B in the case of the one-hot method. The application of natural encoding improves the level of reduction of the system, compared to the one-hot method. The main factor which enables this improvement is the possibility of minimizing such encoded terms and the representation of a larger amount of data by means of the source Π_B , using the same free CM resources as in the case of the one-hot method.

References

- [1] Łuba, T. (2004) Synteza Układów Logicznych. Warsaw University of Technology Press, Warsaw.
- [2] Baranov, S. (1994) Logic Synthesis for Control Automata. Kluwer Academic Publishers, Israel. <http://dx.doi.org/10.1007/978-1-4615-2692-6>
- [3] DeMicheli, G. (1994) Synthesis and Optimization of Digital Circuits. McGraw-Hill, Boston.
- [4] Milik, A. and Hryniewicz, E. (2012) Synthesis and Implementation of Reconfigurable PLC on FPGA Platform. *International Journal of Electronics and Telecommunications*, **58**, 85-94. <http://dx.doi.org/10.2478/v10177-012-0012-8>
- [5] Barkalov, A. and Węgrzyn, M. (2006) Design of Control Units with Programmable Logic. University of Zielona Góra Press, Zielona Góra.
- [6] Klimowicz, A. and Salauyou, V. (2012) The Synthesis of Combined Mealy and Moore Machines Structural Model Using Values of Output Variables as Codes of States. *15th Euromicro Conference on Digital System Design (DCD)*, Izmir, 5-8 September 2012, 789-794. <http://dx.doi.org/10.1109/DSD.2012.130>
- [7] Kubica, M. and Kania, D. (2011) Synteza Logiczna Zespołu Funkcji Ukierunkowana na Minimalizację Liczby Wykorzystywanych Bloków Logicznych PAL w Oparciu o Zmodyfikowany Graf Wyjść. *Pomiary, Automatyka, Kontrola*, **57**, 737-740.
- [8] Altera (2009) Max II Device Handbook. http://www.altera.com/literature/hb/max2/max2_mii5v1.pdf
- [9] Czerwinski, R. and Kania, D. (2012) Area and Speed Oriented Synthesis of FSMs for PAL-based CPLDs. *Microprocessors & Microsystems*, **36**, 56-61. <http://dx.doi.org/10.1016/j.micpro.2011.06.004>
- [10] Salauyou, V. and Grzes, T. (2007) FSM State Assignment Methods for Low-Power Design. *6th International Conference on Computer Information Systems and Industrial Management Applications*, Minneapolis, 28-30 June 2007, 345-350. <http://dx.doi.org/10.1109/CISIM.2007.32>
- [11] Barkalov, A. and Titarenko, L. (2008) Logic Synthesis for Compositional Microprogram Control Units. Springer. <http://dx.doi.org/10.1007/978-3-540-69285-0>
- [12] Altera (2007) Using the UFM in MAX II Devices. www.altera.com/literature/an/an489.pdf
- [13] Cypress (2003) Delta39K ISR CPLD Family. <http://pdf.datasheetcatalog.com/datasheet2/9/Opfaevx4ushkk0zzjksaycghqky.pdf>
- [14] Barkalov, A., Titarenko, L. and Smoliński, L. (2013) Hardware Reduction for Compositional Microprogram Control Unit Dedicated for CPLD Systems. *Proceedings of IEEE East-West Design and Test Symposium EWDTs'13*, Rostov-on-Don, 27-30 September 2013, 1-6. <http://dx.doi.org/10.1109/EWDTs.2013.6673200>

- [15] Barkalov, A., Titarenko, L. and Smoliński, L. (2014) CMCU Model with Base Structure Dedicated for CPLD Systems. *Przegląd Elektroniczny*. (in Press)
- [16] Baranov, S. (2008) Logic and System Design of Digital Systems. TUT Press, TelAviv.
- [17] Adamski, M. and Barkalov, A. (2006) Architectural and Sequential Synthesis of Digital Devices. University of Zielona Góra Press, Zielona Góra.
- [18] Wisniewska, M., Wisniewski, R. and Adamski, M. (2009) Reduction of the Microinstruction Length in the Designing Process of Microprogrammed Controllers. *Przegląd Elektroniczny*, **11**, 114-117.
- [19] Barkalov, A., Titarenko, L. and Smoliński, L. (2012) Optimization of Control Unit based on Construction of CPLD. *Pomiary, Automatyka, Kontrola*, **58**, 93-96.
- [20] Barkalov, A., Titarenko, L. and Smoliński, L. (2011) Optimization of Microprogram Control Unit with Code Sharing. *Proceedings of IEEE East-West Design and Test Symposium (EWDTS)*, Sevastopol, 9-12 September 2011, 55-59. <http://dx.doi.org/10.1109/EWDTS.2011.6116573>
- [21] Maxfield, C. (2004) The Design Warrior's Guide to FPGAs: Devices, Tools and Flows. Elsevier.
- [22] Wiśniewski, R. (2009) Synthesis of Compositional Microprogram Control Units for Programmable Devices. University of Zielona Góra Press, Zielona Góra.
- [23] Barkalov, A. and Titarenko, L. (2009) Logic Synthesis for FSM-Based Control Units. Springer, Zielona Góra. <http://dx.doi.org/10.1007/978-3-642-04309-3>
- [24] Kołopieńczyk, M. (2008) Application of Address Converter for Decreasing Memory Size of Compositional Microprogram Control Unit with Code Sharing. *Lecture Notes in Control and Computer Science*, **12**, 88-96.
- [25] Bieganowski, J. (2011) Synthesis of Microprogram Control Units Oriented toward Decreasing the Number of Macrocells of Addressing Circuit. *Lecture Notes in Control and Computer Science*, **17**, 103-109.
- [26] Salauyou, V. and Klimowicz, A. (2010) Synteza logiczna układów cyfrowych w strukturach programowalnych. Oficyna Wydawnicza Politechniki Białostockiej, Białystok.

Information

Mr. Łukasz Smoliński is a scholar within Sub-measure 8.2.2 Regional Innovation Strategies, Measure 8.2 Transfer of knowledge, Priority VIII Regional human resources for the economy Human Capital Operational Programme co-financed by European Social Fund and state budget.



Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or [Online Submission Portal](#).

