

Alexander BARKALOV, Larysa TITARENKO, Jacek BIEGANOWSKI

INSTYTUT INFORMATYKI I ELEKTRONIKI
ul. Licealna 9, 65-417 Zielona Góra

Kodowanie klas POLC w mikroprogramowanych układach sterujących

Prof. dr hab. inż. Alexander BARKALOV

Prof. Alexander A. Barkalov w latach 1976-1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki Donieckiej. Współpracował aktywnie z Instytutem Cybernetyki im. V.M. Glushkova w Kijowie, gdzie uzyskał tytuł doktora habilitowanego ze specjalnością informatyka. W latach 1996-2003 pracował jako profesor w Instytucie Informatyki Narodowej Politechniki Donieckiej. Od 2004 pracuje jako profesor na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: A.Barkalov@iie.uz.zgora



Mgr inż. Jacek BIEGANOWSKI

Ukończył w roku 2003 studia na kierunku informatyka o specjalności inżynieria komputerowa. Od 2004 roku pracuje w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego na stanowisku asystenta. Jego zainteresowania związane są z programowaniem, systemami operacyjnymi, techniką cyfrową oraz algorytmami ewolucyjnymi.

e-mail: j.bieganowski@iie.uz.zgora.pl



Dr hab. inż. Larysa TITARENKO

Dr hab. Larysa Titarenko w 2004 roku obroniła rozprawę habilitacyjną i uzyskała tytuł doktora habilitowanego ze specjalnością telekomunikacja. W latach 2004-2005 pracowała jako profesor w Narodowym Uniwersytecie Radioelektroniki w Charkowie. Od 2005 pracuje na Wydziale Elektro-techniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: L.Titarenko@iie.uz.zgora



1. Wprowadzenie

Bardzo często system cyfrowy zawiera układ sterujący, który jest odpowiedzialny za koordynowanie współpracy pozostałych elementów tego systemu [10]. Dobór odpowiedniej struktury układu sterującego zależy w bardzo dużym stopniu od implementowanego algorytmu sterowania [3]. Jeżeli liczba operacji (wierzchołków operacyjnych) jest co najmniej dwa razy większa od liczby łańcuchów bloków operacyjnych (OLC) wtedy warto rozważyć zastosowanie mikroprogramowanego układu sterującego [1].

Układy programowalne FPGA są obecnie szeroko stosowane do implementacji układów sterujących [10, 12]. Układy FPGA charakteryzują się dużą liczbą konfigurowalnych bloków logicznych, w których funkcje logiczne realizowane są z użyciem tablic LUT o ograniczonej liczbie wejść (zazwyczaj nie więcej niż 6) [16, 17]. Ograniczona liczba wejść tablic LUT powoduje potrzebę stosowania dekompozycji funkcjonalnej, która może powodować spowolnienie układu ze względu na dużą liczbę poziomów potrzebnych do realizacji części kombinacyjnej układu sterującego (opóźnienie wprowadzają tablice LUT oraz programowalne połączenia między nimi) [13]. Liczba poziomów potrzebnych do realizacji części kombinacyjnej układu sterującego może zostać zmniejszona dzięki wykorzystaniu osadzonych bloków pamięci (EMB), które stają się obecnie standardowym elementem składowym większości produkowanych układów FPGA [7]. Rozwiązanie takie można również zastosować w przypadku mikroprogramowanych układów sterujących.

Redukcja zasobów sprzętowych układu sterującego jest obecnie ważnym zagadnieniem [8, 10]. Zmniejszenie zasobów sprzętowych potrzebnych do realizacji układu sterującego pozwala zmniejszyć między innymi koszty produkcji systemu cyfrowego, zużycie energii, czy też liczbę elementów potrzebnych do realizacji systemu [11]. W przypadku układów FPGA zasoby sprzętowe potrzebne do realizacji układu sterującego można zredukować dzięki uproszczeniu (zmniejszeniu liczby zmiennych) w funkcjach boolowskich opisujących część adresową układu.

2. Sieć działań

Jednym ze sposobów reprezentacji algorytmu sterowania jest sieć działań (ang. *Graph Scheme of Algorithm* – GSA). Przykładowa sieć działań została przedstawiona na rysunku 1.

Sieć działań Γ składa się ze zbioru wierzchołków $V = v_0, v_E \cup V_M \cup V_C$ oraz zbioru krawędzi E . Wierzchołek v_0 jest wierzchołkiem początkowym, v_E jest wierzchołkiem końcowym, V_M jest zbiorem wierzchołków operacyjnych, V_C jest zbiorem wierzchołków warunkowych. Wierzchołek operacyjny $v_q \in V_M$ zawiera mikroinstrukcję $Y(v_q) \in Y$ będącą kolekcją mikrooperacji ze zbioru $Y = \{y_1, \dots, y_N\}$. Wierzchołek warunkowy $v_q \in V_C$ reprezentuje jeden z elementów $x_e \in X$ ze zbioru warunków $X = \{x_1, \dots, x_L\}$.

Streszczenie

W artykule przedstawiono rezultaty syntezy sześciu struktur układów mikroprogramowanych (CMCU), które wykorzystują koncepcję podziału zbioru łańcuchów operacyjnych na klasy łańcuchów pseudorównoważnych (POLC). Przedstawione w pracy struktury układów mikroprogramowanych są przeznaczone przede wszystkim do zastosowania w układach FPGA. Część kombinacyjna układu mikroprogramowanego jest realizowana z użyciem tablic LUT, natomiast pamięć sterująca jest implementowana z użyciem osadzonych bloków pamięci. Badania przeprowadzono dla czterech popularnych kodowań stanów: kodowania binarnego, kodowania one-hot, kodowania Gray'a oraz kodowania Johnson'a.

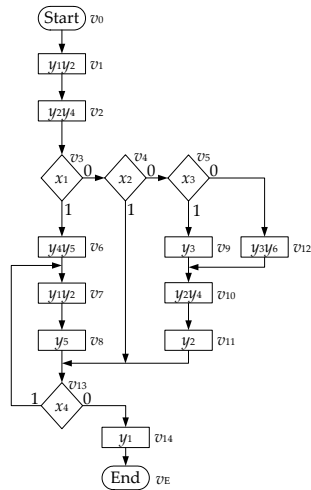
Słowa kluczowe: układ mikroprogramowany, współdzielenie kodów, łańcuch bloków operacyjnych, układ FPGA, tablica LUT, osadzony blok pamięci.

The encoding of POLC classes in microprogram control units

Abstract

The paper presents new synthesis results of six structures of a compositional microprogram control unit (CMCU) targeted mainly at FPGAs. The structure of CMCU consist of two main parts: a control memory and an addressing circuit. The control memory stores microinstructions which are sent to the data path. The addressing circuit is responsible for selecting a microinstruction from the control memory. The addressing part of the CMCU is implemented using LUT tables, while the control memory is implemented using embedded memory blocks (EMB). Partitioning the set of operational linear chains (OLC) into pseudoequivalent classes of chains (POLC) is used in all structures to reduce the size of the CMCU addressing part. The codes of POLCs are stored in the control memory by extending the microinstruction format or by inserting additional control microinstructions (Figs. 2, 3 and 4). The CMCU structures were tested using linear graph-schemes of the algorithm (see Tab. 1). The synthesis was made in Xilinx ISE and Altera Quartus for FPGA and CPLD devices. The synthesis results (Figs. 5 and 6) show that the size of the combinational part for the tested CMCU structures can be reduced by 20% to 50% depending on the CMCU structure (when compared to the base structure – average results). The results also show that the natural binary encoding and Gray's encoding are best for POLC classes. Both encodings give the smallest size of the addressing part and require less control memory space.

Keywords: microprogram control unit, code sharing, FPGA, LUT, embedded memory.



Rys. 1. Sieć działań
Fig. 1. Graph scheme of the algorithm

Niech zbiór $C_{OLC} = \{a_1, \dots, a_i\}$ będzie zbiorem łańcuchów operacyjnych w sieci działań Γ . Łańcuchem operacyjnym $a_i \in C_{OLC}$ nazywa się sekwencję kolejnych wierzchołków operacyjnych, w której dla każdej pary wierzchołków istnieje łuk należący do zbioru E . Łańcuch OLC ma tylko jedno wejście i skończoną liczbę wyjść [1].

Niech zbiór $C_{EOLC} = \{a_1, \dots, a_i\}$ będzie zbiorem elementarnych łańcuchów operacyjnych w sieci działań Γ . Elementarnym łańcuchem operacyjnym nazywa się sekwencję kolejnych wierzchołków operacyjnych, w której dla każdej pary wierzchołków istnieje łuk należący do zbioru E . Łańcuch EOLC ma dokładnie jedno wejście i wyjście [1].

Liniovą siecią działań nazywa się taką sieć działań, w której liczba wierzchołków operacyjnych jest co najmniej dwa razy większa od liczby łańcuchów OLC.

Niech $\Pi = \{V_1, \dots, V_i\}$ będzie podziałem zbioru C na klasy łańcuchów pseudorównoważnych $V_i \in \Pi$. Dwa łańcuchy a_i, a_k są pseudorównoważne, jeżeli ich wyjścia są połączone z wejściem tego samego wierzchołka [1, 3].

Jako przykład rozważmy sieć działań Γ_1 przedstawioną na rysunku 1. Dla sieci działań Γ_1 można wyznaczyć: zbiór warunków logicznych $X = \{x_1, x_2, \dots, x_4\}$, zbiór mikrooperacji $Y = \{y_1, y_2, \dots, y_6\}$, zbiór wierzchołków operacyjnych $V_M = \{v_1, v_2, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{14}\}$, zbiór wierzchołków warunkowych $V_C = \{v_3, v_4, v_5, v_{13}\}$, zbiór łańcuchów OLC $COLC = \{a_1, a_2, \dots, a_5\}$, gdzie $a_1 = \{v_1, v_2\}$, $a_2 = \{v_6\}$, $a_3 = \{v_7, v_8\}$, $a_4 = \{v_9\}$, $a_5 = \{v_{10}, v_{11}\}$, $a_6 = \{v_{12}\}$, $a_7 = \{v_{14}\}$ oraz podział zbioru C_{OLC} na klasy łańcuchów pseudorównoważnych $\Pi = \{\{a_4\}, \{a_2, a_3\}, \{a_1, a_5\}\}$, zbiór łańcuchów EOLC $C_{EOLC} = \{a_1, a_2, \dots, a_7\}$, gdzie $a_1 = \{v_1, v_2\}$, $a_2 = \{v_6\}$, $a_3 = \{v_7, v_8\}$, $a_4 = \{v_9\}$, $a_5 = \{v_{10}, v_{11}\}$, $a_6 = \{v_{12}\}$, $a_7 = \{v_{14}\}$ oraz podział zbioru C_{EOLC} na klasy łańcuchów pseudorównoważnych $\Pi = \{\{a_1\}, \{a_2\}, \{a_3, a_5\}, \{a_4, a_6\}\}$.

3. Mikroprogramowany układ sterujący

Mikroprogramowany układ sterujący składa się z dwóch zasadniczych części: układu adresowego oraz pamięci sterującej. Zadaniem układu adresowego jest wytworzenie adresu mikroinstrukcji dla pamięci sterującej. W pamięci sterującej umieszczone są mikrooperacje wyprowadzane na wyjście układu oraz zmienne sterujące z_i wykorzystywane do sterowania pracą części adresowej. Na rysunkach 2, 3, i 4 przedstawione zostały układy sterujące różniące się częścią adresową.

W układzie mikroprogramowanym ze wspólną pamięcią oraz rozszerzonym formatem mikroinstrukcji (CMCU-CME) (rys. 2) układ adresowy składa się z układu kombinacyjnego oraz licznika. Jeżeli $z_1=1$, wtedy zawartość licznika jest zwiększana, w przeciwnym wypadku nowa wartość licznika jest ustalana przez układ kombinacyjny. Pierwszy wariant odpowiada przejściu w ramach

łańcucha OLC. Drugi wariant odpowiada przejściu z wyjścia łańcucha. W układzie CMCU-CME układ kombinacyjny realizuje funkcje:

$$Q_j^{n+1} = \Phi(x_1, x_2, \dots, x_r, z_2^n, z_3^n, \dots, z_w^n) \quad (1)$$

gdzie $j=1, \dots, m$.

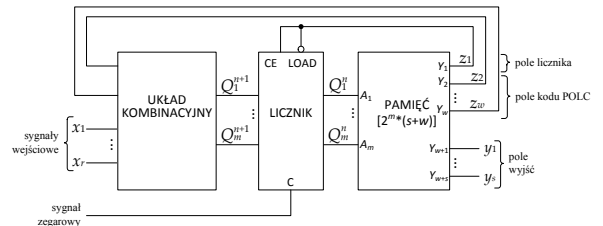
W układzie mikroprogramowanym ze współdzieleniem kodów oraz rozszerzonym formatem mikroinstrukcji (CMCU-CSE) (rys. 3) układ adresowy składa się z układu kombinacyjnego, licznika oraz rejestru. Jeżeli $z_1=1$, wtedy wartość licznika jest inkrementowana, w przeciwnym wypadku nowa wartość licznika oraz zawartość rejestru są generowane przez układ kombinacyjny. W układzie CMCU-CSE układ kombinacyjny realizuje funkcje:

$$P_i^{n+1} = \Psi(x_1, x_2, \dots, x_r, z_2^n, z_3^n, \dots, z_w^n)$$

$$Q_j^{n+1} = \Phi(x_1, x_2, \dots, x_r, z_2^n, z_3^n, \dots, z_w^n)$$

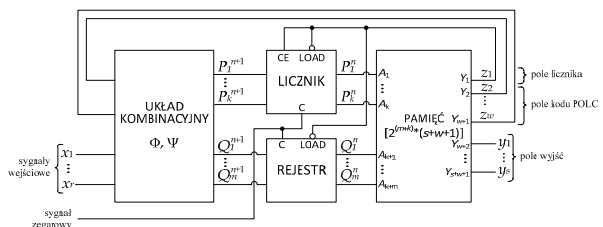
gdzie $i=1, \dots, k; j=1, \dots, m$.

W układzie mikroprogramowanym z elementarnymi łańcuchami oraz rozszerzonym formatem mikroinstrukcji (CMCU-ECE) układ adresowy jest bardzo podobny do układu CMCU-CSE – z tą różnicą, że do adresowania mikroinstrukcji stosuje się łańcuchy EOLC zamiast łańcuchów OLC. Dzięki zastosowaniu łańcuchów EOLC, które posiadają tylko jedno wejście, możliwe jest uproszczenie licznika – gdy zmienna $z_1=1$ licznik jest zerowany. W układzie CMCU-ECE układ kombinacyjny realizuje funkcje (1).



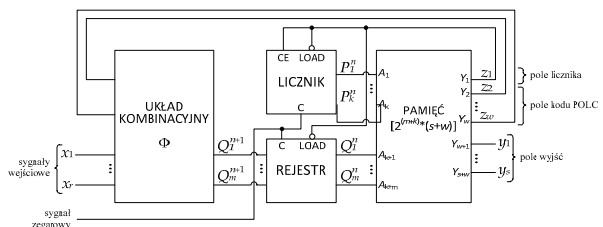
Rys. 2. Mikroprogramowany układ sterujący ze wspólną pamięcią oraz rozszerzonym formatem mikroinstrukcji

Fig. 2. Compositional microprogram control unit with common memory and extended microinstruction format



Rys. 3. Mikroprogramowany układ sterujący ze współdzieleniem kodów oraz rozszerzonym formatem mikroinstrukcji

Fig. 3. Compositional microprogram control unit with code sharing and extended microinstruction format



Rys. 4. Mikroprogramowany układ sterujący z elementarnymi łańcuchami oraz rozszerzonym formatem mikroinstrukcji

Fig. 4. Compositional microprogram control unit with elementary chains and extended microinstruction format

Organizacja pamięci sterującej jest inna dla każdej z przedstawionych struktur, jednak mają one jedną wspólną cechę, a mianowicie we wszystkich trzech strukturach kody klas POLC są umieszczone w pamięci sterującej jako część mikroinstrukcji (zmienne z_2, \dots, z_w). Ten sposób przechowywania kodów POLC jest możliwy do zastosowania, gdy długość słowa osadzonego bloku pamięci implementującego pamięć jest wystarczająca do pomieszczenia dłuższej mikroinstrukcji. Szczegółowe informacje na temat przedstawionych metod syntezy można znaleźć w [6] dla układu CMCU-CME, w [4] dla układu CMCU-CSE oraz w [14] dla układu CMCU-ECE.

Istnieje jeszcze jedna metoda umieszczenia klas POLC w pamięci sterującej – jako dodatkowe mikroinstrukcje sterujące. To podejście stosowane jest w układzie mikroprogramowanym ze wspólną pamięcią oraz mikroinstrukcjami sterującymi (CMCU-CMC), w układzie mikroprogramowanym ze współdzieleniem kodów oraz mikroinstrukcjami sterującymi (CMCU-CSC) oraz w układzie mikroprogramowanym z elementarnymi łańcuchami oraz mikroinstrukcjami sterującymi (CMCU-ECC). W tych strukturach układy adresowe są bardzo podobne do układów adresowych w układach CMCU-CME, CMCU-CSE i CMCU-ECE z tą różnicą, że w inny sposób rozmieszczone są mikroinstrukcje w pamięci sterującej. Szczegółowe informacje o metodach z mikroinstrukcjami sterującymi można znaleźć w [5] dla układu CMCU-CMC, w [4] dla układu CMCU-CSC oraz w [14] dla układu CMCU-ECC.

4. Wyniki eksperymentów

Przedstawione w artykule struktury układów mikroprogramowanych zostały przetestowane z użyciem liniowych sieci działań przygotowanych w ramach pracy [9]. Najważniejsze parametry sieci działań wykorzystanych w badaniach przedstawione zostały w tabeli 1.

Tab. 1. Sieci działań wykorzystane podczas badań
Tab. 1. Graph schemes of the algorithm used for tests

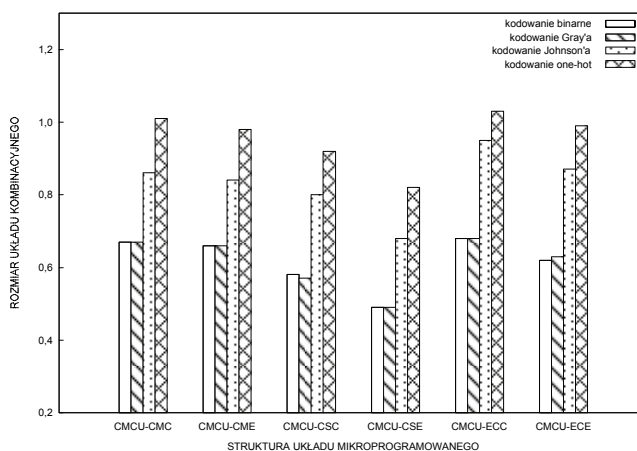
Sieć działań	X	Y	V_M	OLC			EOLC		
				C	C_{max}	Π	C	C_{max}	Π
mk01	10	7	86	14	15	8	19	15	13
mk02	10	7	90	15	10	8	19	10	12
mk03	7	8	49	11	8	6	14	8	9
mk04	9	8	57	14	10	8	18	10	12
mk05	10	9	55	13	9	6	17	6	10
mk06	8	11	59	14	7	11	18	7	15
mk07	11	8	71	17	8	10	22	8	15
mk08	7	7	50	12	9	8	15	7	11
mk09	7	7	57	13	9	11	18	7	16
mk10	13	11	93	21	7	12	26	6	17
mk11	8	8	49	13	6	7	16	5	10
mk12	11	8	72	19	7	13	25	6	19
mk13	6	7	38	9	6	8	12	5	11
mk14	8	7	54	15	6	10	19	5	14
mk15	8	10	69	15	8	12	20	8	17
mk16	6	10	72	12	11	8	15	8	11
mk17	9	7	73	15	8	10	20	7	15
mk18	6	11	52	13	8	10	17	6	14
mk19	6	7	47	12	6	7	14	6	9

X – liczba warunków logicznych, Y – liczba mikrooperacji, V_M – liczba mikroinstrukcji, C – liczba łańcuchów, C_{max} – maksymalna długość łańcucha OLC, Π – liczba klas POLC

Na rysunku 5 przedstawione zostało porównanie średniego rozmiaru części kombinacyjnej układu mikroprogramowanego dla różnych struktur CMCU oraz kodowań klas POLC. Wyniki dla

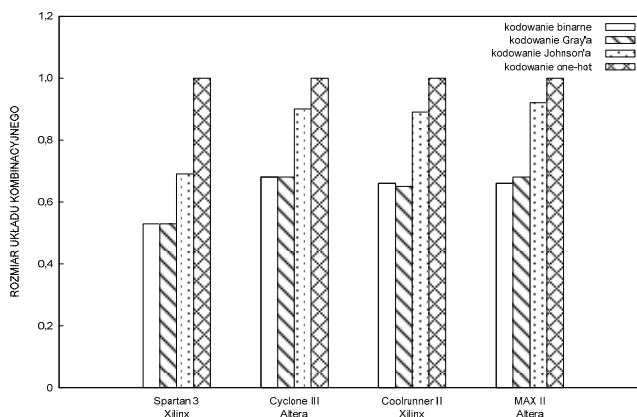
poszczególnych struktur uzyskano porównując średni rozmiar układu adresowego (średnia arytmetyczna dla wszystkich sieci z tab. 1) do rozmiaru układu adresowego w strukturze bazowej. Na przykład dla struktur CMCU-CMC i CMCU-CME układy adresowe porównano do układu adresowego w strukturze CMCU-CM. Wyniki pokazują, że kodowanie binarne i kodowanie Gray'a pozwala uzyskać mniejszą część kombinacyjną niż w przypadku kodowania Johnson'a oraz kodowania one-hot.

Na rysunku 6 przedstawiony został średni rozmiar części kombinacyjnej układu mikroprogramowanego dla różnych układów programowanych. Na rys. 6 widać wyraźnie, że uzyskane wyniki są podobne dla wszystkich testowanych układów programowanych. Oznacza, to że przedstawione w artykule metody syntezy mogą być z powodzeniem stosowane również w przypadku układów CPLD. Należy jednak pamiętać, że większość układów CPLD nie posiada osadzonych bloków pamięci i pamięć sterującą układu mikroprogramowanego trzeba zrealizować z użyciem dodatkowego układu.



Rys. 5. Rozmiar części adresowej układu mikroprogramowanego w zależności od struktury układu oraz kodowania klas POLC

Fig. 5. The size of the addressing circuit depending on the CMCU structure and POLC encoding



Rys. 6. Rozmiar części adresowej układu mikroprogramowanego w zależności od typu układu programowanego oraz kodowania klas POLC

Fig. 6. The size of the addressing circuit depending on the type of programmable device and POLC encoding

5. Podsumowanie

Metody syntezy wykorzystujące podział łańcuchów OLC/EOLC na klasy łańcuchów pseudorównoważnych pozwalają na redukcję liczby tablic LUT w części kombinacyjnej układu mikroprogramowanego implementowanego z użyciem układów FPGA. Wprowadzenie podziału na klasy POLC jest korzystne ponieważ pozwala na zredukowanie liczby zmiennych oraz liczby

wierszy w tablicy przejść układu. Dodatkowo umieszczenie kodów klas POLC w pamięci sterującej pozwala na zastosowanie dowolnej metody kodowania stanów.

W artykule przedstawione zostały rezultaty syntezy sześciu struktur układów mikroprogramowanych oraz czterech metod kodowania stanów. Wyniki eksperymentów pokazują, że kodowanie binarne oraz kodowanie Gray'a okazały się najskuteczniejszymi metodami kodowania spośród badanych. Dłuższe kody klas POLC uzyskane za pomocą kodowania one-hot oraz kodowania Johnson'a prowadziły do wyraźnego wzrostu rozmiaru części kombinacyjnej układu mikroprogramowanego. Oznacza to, że redukcja liczby sygnałów wejściowych w części kombinacyjnej przekłada się na redukcję liczby tablic LUT. W dalszych badaniach planowane jest sprawdzenie bardziej zaawansowanych metod kodowania klas POLC.

6. Literatura

- [1] Adamski M., Barkalov A.: Architectural and Sequential Synthesis of Digital Devices. Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, 2006.
- [2] Baranov S.: Logic and System Design of Digital Systems, TUT Press, 2008.
- [3] Barkalov A., Titarenko L.: Logic synthesis for Compositional Microprogram Control Units, Springer, 2008.
- [4] Barkalov A., Titarenko L., Bieganowski J.: Reduction in the number of LUT elements for control units with code sparing, Inter. Journal of Applied Mathematics and Computer Science, 2010, Vol. 20, Nr. 4.
- [5] Barkalov A., Titarenko L., Bieganowski J.: Synthesis of microprogram control unit with control microinstructions, DES-Des'09: preprints of the 4th IFAC Workshop, 2009.
- [6] Barkalov A., Titarenko L., Bieganowski J.: Synthesis of compositional microprogram control unit with extended microinstruction format, Mixed Design of Integrated Circuits and Systems – MIXDES'2009: 16th Inter. Conf., 2009.
- [7] Borowik G., Fialkowski B., Łuba T.: Cost-efficient synthesis for sequential circuits implemented using embedded memory blocks of FPGA's, IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems, 2007.
- [8] Kam T., Villa T., Brayton R., Sangiovanni-Vincentelli A.: A synthesis of Finite State Machines: Functional Optimization, Kluwer Academic Publishers, Boston, 1998.
- [9] Kołopińczyk M.: Application of address converter for decreasing memory size of compositional microprogram control unit with code sharing, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, 2008.
- [10] Maxfield C.: The Design Warrior's Guide to FPGAs, Academic Press, Inc., Orlando FL, USA, 2004.
- [11] Michali G. D.: Synthesis and Optimization of Digital Circuits, McGraw-Hill, 1994.
- [12] Navabi Z.: Embedded Core Design with FPGAs, McGraw-Hill, 2007.
- [13] Scholl C.: Functional Decomposition with Application of FPGA Synthesis, Kluwer Academic Publishers, 2001.
- [14] Titarenko L., Bieganowski J.: Optimization of compositional microprogram control unit by modification of microinstruction format, Electronics and Telecommunications Quarterly, Vol. 55, Nr. 2, 2009.
- [15] Yang S.: Logic Synthesis and optimization benchmarks user guide, Technical report, Microelectronics Center of North Carolina, 1991.
- [16] Altera Corporation Webpage, <http://www.altera.com>
- [17] Xilinx Corporation Webpage, <http://www.xilinx.com>

otrzymano / received: 18.10.2011

przyjęto do druku / accepted: 01.12.2011

artykuł recenzowany / revised paper

INFORMACJE

WALNE ZGROMADZENIE Polskiego Stowarzyszenia Pomiarów Automatyki i Robotyki „POLSPAR”

W dniu 16 grudnia 2011 r. na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej, odbyło się Walne Zgromadzenie Delegatów Stowarzyszenia „POLSPAR”, którego organem jest miesięcznik PAK. Celem Walnego Zgromadzenia było przyjęcie sprawozdań z działalności Zarządu POLSPAR w upływającej kadencji, dyskusja o celach i planach działalności w następnej kadencji, zatwierdzenie zmian w strukturze organizacyjnej oraz wybory nowych władz Stowarzyszenia.

Zgromadzenie przyjęło 35 osób na nowych członków POLSPAR oraz przyjęło uchwałę o utworzeniu Komitetu Termografii i Termometrii w Podczzerwieni.

Sprawozdanie z działalności Zarządu za okres 2008-2011 przedstawił zastępujący Prezes Stowarzyszenia POLSPAR, prof. W. Winiecki. Ponadto sprawozdania przedstawił: przewodniczący poszczególnych Komitetów (Automatyki, Pomiarów, oraz Robotyki); przedstawiciel Stowarzyszenia POLSPAR w IMEKO (prof. R. Morawski), dr J. Barczyk zdał sprawę z działalności finansowej Stowarzyszenia, a prof. T. Missala przedstawił sprawozdanie z działalności Komisji Rewizyjnej oraz analizę sprawozdań Komitetów.

Następnie odbyła się dyskusja. Po jej zakończeniu, na wniosek Przewodniczącego Komisji Rewizyjnej prof. T. Missali Walne Zgromadzenie Stowarzyszenia POLSPAR jednogłośnie przyjęło uchwałę o przyjęciu sprawozdania Zarządu POLSPAR oraz

o udzieleniu absolutorium za kadencję 2008-2011 ustępującemu Zarządowi.

W głosowaniu tajnym do Zarządu POLSPAR zwykłą większością głosów wybrano następujące osoby: A. Turnau, E. Hryniewicz, J. Szpytko, J. Mindykowski, T. Skubis, R. Morawski, K. Mianowski, J. Warczyński, Z. Pilat, P. Pągowski, H. Madura, W. Minkina, K. Duzinkiewicz, J. Kacprzyk, M. Świercz. Zgodnie ze statutem w skład Zarządu wchodzi również przewodniczący Komitetów Automatyki (prof. Z. Kowalczyk), Pomiarów (prof. W. Winiecki), Robotyki (prof. A. Masłowski), oraz Termografii i Termometrii w Podczzerwieni (prof. B. Więcek).

Na prezesa Stowarzyszenia POLSPAR w głosowaniu jawnym Walne Zgromadzenie jednogłośnie wybrało prof. Z. Kowalczyka. Na sekretarza POLSPAR został wybrany dr hab. inż. M. Świercz, na skarbnika dr hab. inż. K. Duzinkiewicz, a do Komisji Rewizyjnej T. Missala (przewodniczący KR), C. Smutnicki, Z. Warsza, K. Gromaszek, oraz J. Barczyk.

Rozszerzona informacja o Walnym Zgromadzeniu POLSPAR znajduje się na stronie internetowej www.pak.info.pl/Aktualności.

*Opracowanie: Zdzisław Kowalczyk
Prezes POLSPAR*