Larysa TITARENKO, Olena HEBDA, **Alexander BARKALOV**
UNIWERSYTET ZIELONOGÓRSKI, Wydział Elektrotechniki, Informatyki i Telekomunikacji,
ul. Licealna 9, 65-417 Zielona Góra

Redukcja układu logicznego mikroprogramowanego automatu Moore'a przy kodowaniu zbiorów wyjściowych zmiennych

Dr hab. 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 jako adiunkt na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: l.titarenko@iie.uz.zgora.pl

Mgr inż. Olena HEBDA

Mgr inż. Olena Hebda – absolwentka Narodowego Aerokosmicznego Uniwersytetu „KhAI". Ukończyła w roku 2009 studia o specjalności biotechniczne i medyczne aparaty i systemy. Od 2009 roku jest doktorantką na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: O.Shapoval@weit.uz.zgora.pl

Streszczenie

W artykule została przedstawiona metoda syntezy mikroprogramowanego automatu Moore'a implementowanego w układach nano-PLA. Metoda ta jest ukierunkowana na redukcję zasobów sprzętowych, potrzebnych do implementacji automatu Moore'a. Jest ona oparta na przedstawieniu następnego kodu stanu jako konkatenacji kodu klasy zbioru wyjściowych zmiennych i kodu wierzchołka. Takie podejście pozwala wyeliminować zależność między stanami i wyjściowymi zmiennymi, a także zmniejszyć liczbę linii w tabeli przejść automatu Moore'a do odpowiedniej liczby linii w równoważnym automacie Mealy'ego.

Słowa kluczowe: mikroprogramowany automat Moore'a, nano-PLA, stany pseudoekwiwalentne, układ logiczny.

Reduction of a microprogrammable Moore automaton logic circuit with encoding the sets of output variables

Abstract

The model of a microprogrammable Moore automaton is often used during the digital control systems realization [1 – 3]. The development of microelectronics has led to appearance of different programmable logic devices [7, 8] which are used for implementing microprogrammable automaton (MPA) logic circuits. One of the important problems of Moore MPA synthesis is the decrease of chip space occupied by the MPA logic circuit. The methods of solution of this problem depend strongly on logic elements used for implementing the MPA logic circuit [2 – 4]. In this paper we discuss the case when nanoelectronic programmable logic arrays (PLA) are used for implementing the Moore MPA logic circuit. The proposed method is based on representation of the next state code as a concatenation of code for the class of collection of output variables and code of the vertex (Fig. 2). In this method the classes of the pseudoequivalent states are used [1, 9]. Such an approach allows reducing the number of rows of the Moore MPA structure table up to the value of the equivalent Mealy MPA. As a result the area of the matrices generating input memory functions is optimized.

Keywords: microprogrammable Moore automaton, nano-PLA, pseudoequivalent states, logic circuit.

Prof. dr hab. inż. Alexander BARKALOV

Prof. Alexander A. Barkalov w latach 1976-1996 był pracownikiem Instytutu 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.barkalovl@ iie.uz.zgora.pl

1. Wprowadzenie

Jednostka sterująca [1, 2] jest jednym z najważniejszych elementów systemu cyfrowego. Jest odpowiedzialna za zarządzanie innymi blokami projektowanego układu. Jedną z popularnych metod realizacji jednostek sterujących jest zastosowanie modelu mikroprogramowanego automatu (MPA) Moore'a [1]. Jednym z najbardziej istotnych problemów syntezy MPA jest redukcja liczby jednostek sprzętowych, potrzebnych dla implementacji układu logicznego MPA. Jego rozwiązanie polepszy takie parametry jak szybkość działania i zużycie energii [2, 3]. Ponadto może pomóc rozwijającemu się przemysłowi, np. na terenie województwa lubuskiego, ponieważ pozwoli producentom na uzyskanie przewagi konkurencyjnej na rynku w stosunku do innych produktów.

Metody rozwiązania tego problemu w istotny sposób zależą od elementów logicznych wykorzystanych do implementacji układu logicznego MPA [4]. W tym artykule zostanie rozpatrzony przypadek wykorzystania nano-PLA (ang. nanoelectronic programmable logic arrays) dla implementacji MPA Moore'a.

Już w latach 70-tych PLA zostały popularną bazą dla implementacji układów logicznych [5, 6]. Jak zostało zaznaczone w pracy [7], udane struktury obliczeniowe wracają z powrotem w kolejnych etapach rozwoju technologicznego. Obecnie powrót PLA można zaobserwować w hybrydowych FPGA (ang. hybrid Field Programmable Gate Array) [8], w CoolRunner CPLD (ang. Complex Programmable Logic Device) firmy Xilinx oraz we współczesnej nanoelektronice [7]. W nanoelektronice takie urządzenia nazywają się nano-PLA. Aktualnie są prowadzone intensywne badania w dziedzinach związanych z nano-PLA. Największym problemem w układach nanoelektronicznych jest wysokie prawdopodobieństwo wystąpienia wad technologicznych [7]. Dlatego opracowanie nowych metod, które pozwolą zmniejszyć złożoność układu logicznego MPA oraz zmniejszyć liczbę zasobów sprzętowych potrzebnych dla implementacji tego układu, jest bardzo ważne.

2. Klasyczne metody projektowania MPA Moore'a

Automat Moore'a może być przedstawiony jako sieć działań (GSA – ang. Graph Scheme of Algorithm), a także opisany za pomocą tabeli przejść [3]. Tabela przejść składa się z następujących kolumn: a_m, $K(a_m)$, a_s, $K(a_s)$, X_h, Φ_h, . a_m jest początkowym stanem automatu, gdzie $a_m \in A$ jest zbiorem stanów automatu; $K(a_m)$ jest kodem stanu a_m zapisanym na $R = \lceil log_2 M \rceil$ bitach, gdzie M jest liczba stanów automatu (do kodowania stanów wykorzystujemy zmienne wewnętrzne ze zbioru $T = \{T_1, \ldots, T_R\}$). a_s jest następnym stanem automatu; $K(a_s)$ jest kodem stanu a_s. X_h jest koniunkcją zmiennych ze zbioru

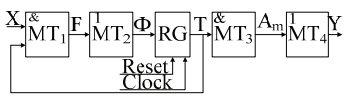
wejściowych warunków cyfrowych $X=\{x_1,\ldots,x_L\}$, wyznaczających przejście $\langle a_m,\ a_s\rangle$. Φ_h jest zbiorem wejściowych funkcji wzbudzeń pamięci, które są równe 1 przy przełączeniu pamięci automatu z kodu $K(a_m)$ na kod $K(a_s)$, $\Phi_h \subseteq \Phi = \{\varphi_1,..,\varphi_R\}$. h jest numerem przejścia automatu, $h=\overline{1,H_1}$. W kolumnie a_m jest wpisany zbiór zmiennych wyjściowych (ZZW) Y_q. Te wyjściowe zmienne są generowane w stanie $a_m \in A$ ($Y_q \subseteq Y=\{y_1,\ldots,y_N\}$, $q=1,\ldots,Q$). Opisana tabela jest wykorzystywana dla tworzenia następujących układów funkcji:

$$\Phi=\Phi(T,X), \tag{1}$$

$$Y=Y(T). \tag{2}$$

Układy funkcji (1) – (2) wyznaczają układ logiczny MPA. Przy wykorzystaniu PLA-bazy układ logiczny MPA Moore'a będzie składać się z dwóch PLA: „centralnego" PLA (CPLA) (ang. Core PLA) i wyjściowego PLA (OPLA) (ang. Output PLA). W tym modelu CPLA implementuje (1), OPLA implementuje układ (2).

Układy funkcji (1) - (2) mogą być zaimplementowane za pomocą tylko dwóch matryc. Pierwsza z nich (AND-matryca) implementuje koniunkcje termów systemów. Druga matryca (OR-matryca) implementuje funkcje. Ale takie rozwiązanie prowadzi do układu MPA z największym możliwym zużyciem zasobów sprzętowych [7]. Znacznie częściej w praktyce używano 4 matryc [1, 2] (rys. 1). Oznaczymy taki model MPA Moore'a jako MPA U_1.



Rys. 1. Matrycowy układ automatu Moore'a U_1
Fig. 1. Matrix implementation of FSM U_1

Przeanalizujmy komponenty układu matrycowego, pokazanego na rys. 1. Koniunkcyjna matryca MT_1 implementuje układ termów $F=\{F_1,\ldots,F_{H_1}\}$; dysjunkcyjna matryca MT_2 – system (1); koniunkcyjna matryca MT_3 – termy A_m ($m=\overline{1,M}$), odpowiednio do stanów MPA; dysjunkcyjna matryca MT_4 – system (2). Rejestr RG przechowuje kody stanów. Sygnał „Start" wykorzystuje się dla załadowania stanu początkowego $a_1 \in A$, a sygnał „Clock" przełącza stan rejestru w zależności od funkcji Φ. Matryce MT_1 i MT_2 wyznaczają blok CPLA, a matryce MT_3 i MT_4 blok OPLA.

Główną wadą automatu Moore'a jest znaczne przekraczanie parametru H_0, który oznacza liczbę wierszy w tabeli przejść równoważnego automatu Mealy'ego. Oprócz tego liczba stanów automatu Moore'a może znacznie przekraczać liczbę stanów M_0 równoważnego automatu Mealy'ego [3]. Dlatego w praktyce często występują następujące warunki:

$$H_0 < H_1;\ R_0=\lceil log_2 M_0 \rceil < R. \tag{3}$$

Złożoność każdej z matryc można określić jako powierzchnię $S(M_i)$ kryształu potrzebną dla jej realizacji $i=\overline{1,4}$. W artykułach teoretycznych tę powierzchnię wyznacza się w jednostkach umownych [1]. Dla automatu U_1 można otrzymać następujące oceny:

$$\begin{aligned} &S(M_1)=2(L+R)H_1;\ S(M_2)=H_1R;\\ &S(M_3)=2R\cdot M;\ \ S(M_4)=M\cdot N. \end{aligned} \tag{4}$$

Powierzchnia $S(U_1)$, którą zajmuje automat U_1, wyznaczana jest jako suma powierzchni (4). Biorąc pod uwagę nierówności (3), powierzchnia zajmowana przez układ logiczny automatu Moore'a zawsze będzie większa od powierzchni potrzebnej dla równoważnego automatu Mealy'ego.

Istnieje wiele metod ukierunkowanych na redukcję zasobów sprzętowych, potrzebnych do realizacji układu logicznego automatu Moore'a [3], ale dotyczą one wykorzystania CPLD lub FPGA technologii i dlatego nie mogą być bezpośrednio użyte w przypadku realizacji MPA Moore'a na bazie PLA. Biorąc to pod uwagę, proponujemy nową metodę syntezy MPA Moore'a przy wykorzystaniu PLA. Jest ona rozwinięciem metody opisanej w[9].

Jedną z cech automatu Moore'a są stany pseudoekwiwalentne [2]. Stany a_i, a_j nazywamy pseudoekwiwalentnymi jeśli odpowiednie do nich operacyjne wierzchołki sieci działań są połączone z wejściem tego samego wierzchołka [3].

3. Podstawowa idea proponowanej metody

Niech $\Pi_A=\{B_1,\ldots,B_I\}$ będzie rozbiciem zbioru stanów na klasy pseudoekwiwalentne. Warto zauważyć, że każda klasa stanów pseudoekwiwalentnych odpowiada unikatowemu stanowi równoważnego automatu Mealy'ego. Zakodujmy klasy $B_i \in \Pi_A$ binarnym kodem $K(B_i)$ mającym $R_B=\lceil log_2 I \rceil$ bitów.

Niech początkowa GSA Γ składa się z Q_0 różnych zbiorów zmiennych wyjściowych (ZZW) $Y_q \subseteq Y$. Zakodujmy zbiór Y_q binarnym kodem $K(Y_q)$ mającym $R_Y=\lceil log_2 Q_0 \rceil$ bitów.

Niech $E=\{b_1,\ldots,b_D\}$ będzie zbiorem wierzchołków operacyjnych GSA Γ. Na podstawie zbioru E utwórzmy nowy zbiór $\Pi_\alpha=\{C_1,\ldots,C_{Q_0}\}$. Do podzbioru C_η wchodzą wierzchołki operacyjne z jednakowym ZZW zapisanym w nich $Y(b_i)=Y(b_j)$ ($i,j \in \{1,\ldots,D\}$). Zakodujmy każdy wierzchołek $b_q \in C_j$ binarnym kodem $K(b_q)$ mającym $R_\alpha=\lceil log_2 G \rceil$ bitów, gdzie $G=max(|C_1|,\ldots,|C_\eta|)$. Użyjmy zmiennych $z_r \in Z_1$ do tego kodowania, $|Z_1|=R_\alpha$. Teraz kod stanu $a_m \in A$ może zostać przedstawiony jako

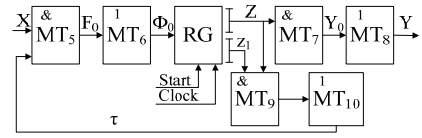
$$K(a_m)=K(Y_q)*K(b_q), \tag{5}$$

gdzie $b_q \in E$ jest wierzchołkiem operacyjnym odpowiadającym stanowi $a_m \in A$, $Y_q=Y(b_q)$ oraz znak * oznacza operacje konkatenacji.

Utwórzmy układ $B=B(A)$, który opisuje zależność między klasami $B_i \in \Pi_A$ i stanami $a_m \in A$. Każda funkcja $B_i \in B$ przedstawiana jest w postaci

$$B_i=\bigvee_{i=1}^{I} C_{im}A_m (i=1,\ldots,I), \tag{6}$$

gdzie symbol C_{im} oznacza zmienne Boolowskie równe 1 jeśli $a_m \in B_i$. Takie podejście prowadzi do MPA Moore'a U_2 (rys. 2).



Rys. 2. Matrycowy układ automatu Moore'a U_2
Fig. 2. Matrix implementation of FSM U_2

W MPA U_2 matryca MT_5 implementuje układ termów F_0, który odpowiada wierszom w tabele przejść i zależy od warunków logicznych $x_l \in X$ i dodatkowych zmiennych $\tau_r \in \tau$, wykorzystywanych dla kodowania klas $B_i \in \Pi_A$, gdzie $|\tau| = R_B$. Matryca MT_6 realizuje wejściowe funkcje pamięci

$$\Phi_0 = \Phi_0(\tau, X). \tag{7}$$

Układ (7) składa się z $R_Y + R_\alpha$ funkcji, co wyznacza liczbę przerzutników w RG. Matryca MT_7 implementujt termy Y_0 wchodzące w układ $y_n \in Y$ i zależne od zmiennych $z_r \in Z$, gdzie $|z| = R_Y$. Matryca MT_8 implementuje funkcje $y_n \in Y$, które zależą od termów $\Delta_q \in Y_0$. Matryca MT_9 realizuje termy A_0 ze wzoru (6) oraz matryca MT_{10} funkcje $\tau_r \in \tau$, które są wykorzystywane do kodowania klas $B_i \in \Pi_A$, gdzie $|\tau| = R_B$.

Matryce MT_5 i MT_6 tworzą blok CPLA oraz matryce MT_7 i MT_8 blok OPLA implementujący funkcje $Y = Y(Z)$.

Matryce MT_9 i MT_{10} tworzą blok transformacji kodów generujący funkcje $\tau = \tau(z, z_1)$.

Złożoność układu logicznego MPA Moore'a U_2 może być wyrażona jako:

$$\begin{aligned} &S(M_5) = 2(L + R_B)H_0;\ S(M_6) = (R_Y + R_\alpha)H_0;\ S(M_7) = 2R_Y q;\\ &S(M_8) = qN;\ S(M_9) = 2(R_Y + R_\alpha)H';\ S(M_{10}) = R_B H'. \end{aligned} \tag{8}$$

4. Analiza proponowanej metody

Porównajmy proponowaną metodę z metodą optymalnego kodowania stanów, która jest jedną z podstawowych metod opartych na istnieniu klas pseudoekwiwalentnych. Przy optymalnem kodowaniu stanów *s*tan $a_m \in A$ jest kodowany w taki sposób, żeby każdą klasę $B_i \in \Pi_A$ przedstawić jednym uogólnionym interwałem R-wymiarowej przestrzeni Boole'a. To prowadzi do automatu U_3, którego schemat pokrywa się ze schematem na rys. 1, ale matryca M_1 realizuje tylko $H_2 = H_0$ termów. Jednak takie kodowanie nie zawsze może być wykonane [2], co prowadzi do tego, że $H_2 > H_0$.

Użyjemy probabilistycznego podejścia zaproponowanego w [10] i opracowanego w [11]. Takie podejście ma trzy kluczowe punkty:

1. Zastosowanie klasy GSA zamiast określonego GSA. Ta klasa charakteryzuję się parametrami p_1 i p_2. Oznaczymy liczbę wierzchołków w GSA Γ przez Q, wierzchołków operacyjnych przez Q_1, a wierzchołków warunkowych przez Q_2. $p_1 = Q_1/(Q-2)$ ($p_2 \approx 1 - p_1$) jest prawdopodobieństwem tego, że okreslony wierzchołek GSA Γ będzie operacyjnym (warunkowym).

2. Zastosowanie matrycowej implementacji układu logicznego jednostki sterującej [1]. Umożliwia to wyznaczenie ilości zasobów sprzętowych jako objętośći matryc danej jednostki sterującej.

3. Zastosowanie charakterystyk względnych zamiast bezwzględnych. Będziemy analizować relacje $\eta = S(U_i)/S(U_j)$, gdzie $S(U_{i,j})$ jest liczbą zasobów sprzętowych potrzebnych do implementacji układu jednostki sterującej.

Skorzystajmy z wyników prac [10, 11], gdzie oszacowanie głównych parametrów MPA przedstawiono jako funkcje charakterystyk GSA i współczynników:

$$\begin{aligned} &H_0 = 4.44 + 1.44 p_1 Q/p_3;\ H = 12.6 + 2.16 p_1 Q/p_3;\\ &M = p_1 Q + 1;\ L = (1 - p_1)Q/p_4; \end{aligned} \tag{9}$$

gdzie $p_3 = p_1 Q/Q_0 \in \{1.1;1.2\}$, $p_4 = p_2 Q/L \in \{1.1;1.2\}$.

Przeanalizujmy metodę optymalnego kodowania stanów. Zużycie zasobów sprzętowych dla MPA U_3 jest zdefiniowane jako

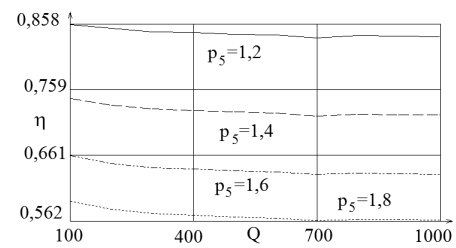
$$S(U_3) = \sum_{i=1,4} S(M_i) = p_5 H_0(3R + 2L) + M(2R + N); \tag{10}$$

gdzie $p_5 = \overline{1{,}2;1{,}8}$ jest współczynnikiem efektywności metody optymalnego kodowania. Używając równania (10), możemy otrzymać $S(U_3) = f(p_1, Q, p_3, p_4, N, p_5)$. Równanie końcowe jest zbyt duże i mało informatywne dla czytelnika, dlatego nie będziemy jego tu zamieszczać.

W podobny sposób możemy otrzymać $S(U_2) = f(p_1, Q, p_3, p_4, N, p_6, p_7)$, $H' = Mp_6$, $p_6 = \overline{0{,}5;0{,}8}$, $G = p_7(G_{max} - G_{min}) + G_{min}$, gdzie $G_{max} = Q_1 - Q_0 + 1$ jest maksymalną teoretycznie możliwą liczba wierzchołków z jednakowym ZZW, $G_{min} = \lceil Q_1/Q_0 \rceil$ jest minimalną możliwą liczbą wierzchołków z jednakowym ZZW, $p_7 = \overline{0{,}1}$.

Analiza różnych GSA pokazała, że $Q \in \{100,\ldots,1000\}$ i $N \le 100$. Przeanalizujmy funkcje $\eta = S(U_2)/S(U_3)$. Przykładowe wyniki zostały pokazane na rys. 3.

Analiza funkcji η pokazała, że zużycie zasobów sprzętowych w modelu U_2 jest niższe niż w U_3. Najlepsze wyniki otrzymujemy dla GSA z liczbą wierzchołków $700 \le Q \le 1000$. Oprócz tego funkcje η w istotny sposób zależą od G. Najlepsze wyniki są osiągane w sytuacji gdy liczba wierzchołków w podklasach C_{Q_0} znacznie nie różni się ($p_7 < 0{,}5$). W tym przypadku oszczędność może wynieść nawet 44%, co jednak zależy i od innych parametrów.



Rys. 3. Porównanie modelu U_2 i U_3: p_1=0,2; p_3=p_4=1,1; N=20; p_6=0,5; p_7=0
Fig. 3. Comparison of models U_2 and U_3: p_1=0,2; p_3=p_4=1,1; N=20; p_6=0,5; p_7=0

5. Wnioski

Proponowana metoda jest ukierunkowana na zmniejszenie liczby zasobów sprzętowych potrzebnych do implementacji układu logicznego MPA Moora. Zaproponowane podejście pozwala zmniejszyć liczbę termów w układzie funkcji wzbudzeń pamięci MPA Moore'a do odpowiedniej wartości w równoważnym automacie Mealy'ego oraz zmniejszyć liczbę termów w układzie wyjściowych zmiennych dzięki eliminacji zależności między kodami stanów i kodami ZZW.

Porównanie proponowanej metody z metodą optymalnego kodowania stanów pokazała, że proponowana metoda pozwala zredukować liczbę zasobów sprzętowych potrzebnych do implementacji układu logicznego MPA Moore'a co najmniej o 10% w 93,7% przypadkach i co najmniej o 35% w 22,5 % przypadkach. Bardzo istotnym czynnikiem wpływającym na efektywność metody jest rozkład ZZW w wierzchołkach operacyjnych. Najlepsze wyniki można otrzymać przy równomiernym rozkładzie ZZW w wierzchołkach operacyjnych, co odpowiada GSA z najmniejszym możliwym znaczeniem parametru G .

Współautorka – Olena Hebda– jest stypendystą w ramach Poddziałania 8.2.2 „Regionalne Strategie Innowacji", Działania 8.2 „Transfer wiedzy", Priorytetu VIII „Regionalne Kadry Gospodarki" Programu Operacyjnego Kapitał Ludzki współfinansowanego ze środków Europejskiego Funduszu Społecznego Unii Europejskiej i z budżetu państwa.

6. Literatura

[1] Baranov S.: Logic synthesis for control automata. Norwell, MA, USA: Kluwer Academic Publishers, 1994.
[2] Barkalov A., Titarenko L.: Logic synthesis for FSM-based control units. Lecture notes in electrical engineering. Berlin: Springer Verlag Heidelberg, 2009, vol. 53.
[3] Baranov S.: Logic and system design of digital systems. Tallinn: TUT Press, 2008.
[4] Kania D., Czerwiński R.: Area and speed oriented synthesis of FSM for PAL-based CPLDs. Microprocessors and Microsystems 2012; 36(1), 45-61.
[5] DeMicheli G., Brayton R., Sangiovanni-Vincentelli A.: Optimal state assignment for finite state machines. IEEE Trans. on CAD 1985; 4, 269 -284.
[6] Levin I.: Decomposition design of automata based on PLA with memory. Automatic Control and Computer Sciences 1986; 20(2), 61-68.
[7] Baranov S., Levin I., Koren O., Karpovsky M.: Designing fault tolerant FSM by nano-PLA. in Proc. of the 15th IEEE International On- Line Testing Symposium; 2009, Sembra-Lisbon, Portugal, 229-234.
[8] Singh S., Singh R., Bhatia M.: Performance evaluation of hybrid reconfigurable computing architecture over symmetrical FPGAs. International Journal of Embedded Systems and Applications 2012; 2(3), 107-116.
[9] Barkalov A., Titarenko L., Hebda O.: Synthesis of Moore FSM with encoding of collections of microoperations implemented with ASIC. Pomiary, Automatyka, Kontrola 2012; 58, nr 6, 514-518.
[10] Novikov G.: About one approach for finite state machines research, Contr. Syst. Mach. 1974; 2, 70–75, (in Russian).
[11] Barkalov A.: Synthesis of Control Units with Programmable Logic Devices. DNTU, Donetsk, 2002, in Russian.

otrzymano / received: 15.05.2013
przyjęto do druku / accepted: 03.07.2013 *artykuł recenzowany / revised paper*

INFORMACJE

Szanowni Autorzy artykułów publikowanych w PAK

W trosce o jak najwyższy poziom punktacji miesięcznika PAK zwracam się z prośbą o cytowanie artykułów opublikowanych w PAK w innych artykułach, zwłaszcza tych publikowanych w czasopismach z listy filadelfijskiej. Ma to bezpośredni wpływ na współczynnik IF (Impact Factor) miesięcznika PAK.

W algorytmach oceny czasopism współczynnik IF ma największą wagę. Na zwiększenie wartości współczynnika IF redakcja czasopisma nie ma żadnego wpływu, ale wszystko zależy od Autorów cytujących. W przypadku miesięcznika PAK aktualnie każde cytowanie zwiększa IF o około 0,002. Oczywiście cytowanie artykułu tylko wtedy jest uzasadnione, jeżeli jest on tematycznie związany z artykułem cytującym, a autor korzystał z niego przy przygotowaniu pracy.

Aby ułatwić Autorom korzystanie z artykułów opublikowanych w PAK (a także możliwość cytowania) została opracowana przez redakcję PAK „Wyszukiwarka", umożliwiająca wyszukiwanie artykułów według nazwiska autora, słowa tytułu artykułu, albo frazy kluczowej.

Aby skorzystać z „Wyszukiwarki" należy:
- wejść na stronę: www.pak.info.pl
- w menu „Wyszukiwarka" (po lewej stronie ekranu) wybrać „Artykuły".

Strona zawiera również szereg innych łatwo dostępnych funkcjonalności, m.in. wykazy artykułów opublikowanych w PAK, a cytowanych w artykułach opublikowanych w czasopismach z listy filadelfijskiej.

Zdaję sobie sprawę, że redakcje niektórych czasopism usuwają cytowania artykułów publikowanych w czasopismach spoza listy filadelfijskiej, np. argumentując, że są one mało dostępne. Taka argumentacja będzie mniej uzasadniona, jeżeli tytuł naszego miesięcznika oraz tytuły artykułów będą podane w cytowaniach w języku angielskim. Proszę zauważyć, że oficjalny tytuł anglojęzyczny miesięcznika PAK (występujący na okładce) ma formę: Measurement, Automation and Monitoring (MA&M), a wszystkie artykuły naukowe publikowane w PAK są napisane albo w języku angielskim, albo mają rozszerzone abstrakty w tym języku.

Tadeusz SKUBIS
Redaktor naczelny Wydawnictwa PAK