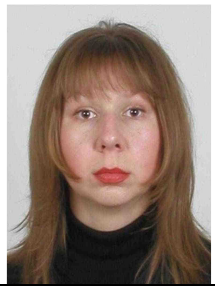
Larysa TITARENKO, Olena HEBDA, **Alexander BARKALOV**
UNIVERSITY OF ZIELONA GORA , Podgórna 50, 65-256, Zielona Góra

Optymalizacja układu logicznego mikroprogramowanego automatu Moore'a przy użyciu nano-PLA

Dr hab. inż. Larysa TITARENKO, prof. UZ



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 Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

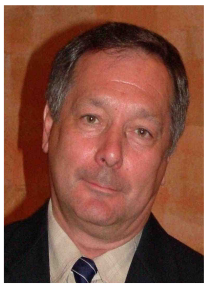
e-mail: L.Titarenko@iie.uz.zgora.pl

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

Prof. dr hab. inż. Alexander BARKALOV, prof. UZ



Prof. Alexander A. Barkalov w latach 1976-1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki w Donieckiej. Współpracował aktywnie z Instytutem Cybernetyki im. V.M. Glushkova w Kijowie, gdzie uzyskal 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.pl

Streszczenie

W artykule została przedstawiona metoda syntezy mikroprogramowanego automatu Moore'a implementowanego w układach nano-PLA. Metoda jest ukierunkowana na redukcję zasobów sprzętowych, potrzebnych do implementacji automatu Moore'a. Jest ona oparta na optymalnym kodowaniu stanów i rozbijaniu matrycy termów na dwie części. Takie podejście pozwala zmniejszyć liczbę linii w tabeli przejść automatu Moore'a do odpowiedniej liczby linii w ekwiwalentnym automacie z wyjściami typu Mealy'ego.

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

Optimization of logic circuit of microprogrammed Moore machine implementing with nano-PLA

Abstract

The method is proposed for reduction of the area of matrix implementation of the circuit of the Moore finite state machine. The method is based on optimal state coding and decomposition of a matrix of terms on two submatrixes. Thus classes of the pseudoequivalent states are used. Such approach allows reducing the number of lines of the table of transitions of Moore FSM up to this value of the equivalent Mealy FSM.

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

1. Wprowadzenie

Jednostka sterująca [3] jest jednym z najważniejszych elementów systemu cyfrowego. Jedną z popularnych metod realizacji jednostek sterujących jest zastosowanie modelu automatu mikroprogramowanego (MPA) Moore'a [3]. 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 [1, 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.

W tym artykule zostanie rozpatrzony wypadek 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. Obecnie powrót PLA można zaobserwować w hybrydowych FPGA [7], w CoolRunner CPLD firmy Xilinx oraz we współczesnej nanoelektronice [2, 6]. Aktualnie są prowadzone intensywne badania w dziedzinach związanych z nano-PLA [2]. Największym problemem w układach nanoelektronicznych jest wysokie prawdopodobieństwo wystąpienia wad technologicznych [2]. 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

Automat Moore'a może być przedstawiony jako sieć działań (GSA) (ang. Graph Sheme of Algorithm), a także opisany za pomocą tabeli przejść [1]. Tabela przejść składa się z następujących kolumn: a_m, $K(a_m)$, a_s, $K(a_s)$, X_h, Φ_h, 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 (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 $a_m \in A$ ($Y_q \subseteq Y = \{y_1, \ldots, y_N\}$, $q = 1, ..., Q$). Opisana tabela jest wykorzystywana dla tworzenia następujących układów funkcji:

$$\Phi = \Phi(T, X), \quad (1)$$

$$Y = Y(T). \quad (2)$$

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 [2]. Znacznie częściej w praktyce używano 4 matryc [3, 4]. Oznaczymy taki model MPA Moore'a jako MPA U_1.

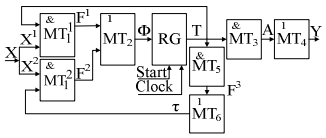
Główną wadą automatu Moore'a jest znaczne przekraczanie parametru H_0, który oznacza liczbę wierszy w tabeli przejść ekwiwalentnego automatu Mealy'ego. Oprócz tego liczba stanów automatu Moore'a może znacznie przekraczać liczbę stanów M_0 ekwiwalentnego automatu Mealy'ego [1].

Jedną z cech automatu Moore'a są stany pseudoekwiwalentne [3]. Stany a_i, a_j nazywamy pseudoekwiwalentnymi jeśli odpo-

wiednie do nich operatorowe wierzchołki sieci działań są połączone z wejściem tego samego wierzchołka [1]. Są dwie podstawowe metody oparte na istnieniu klas pseudoekwiwalentnych [3]: metoda optymalnego kodowania stanów (U_2) i metoda transformacji kodów stanów (U_3).

3. Podstawowa idea proponowanej metody

Niech $\Pi_A = \{B_1,\dots,B_I\}$ będzie rozbiciem zbioru stanów na klasy pseudoekwiwalentne. Zakodujmy stany $a_m \in A$ w taki sposób, żeby maksymalnie możliwa liczba klas $B_i \in \Pi_A$ była przedstawiona jednym interwałem R-wymiarowej przestrzeni Boole'a. Odniesiemy takie klasy do podzbioru Π_{RG}, a resztę do podzbioru Π_{CT}, gdzie $|\Pi_{CT}| = I_{CT}$. Zakodujmy klasy $B_i \in \Pi_{CT}$ za pomocą binarnego kodu $K(B_i)$, gdzie $R_{CT} = \lceil log_2(I_{CT}+1) \rceil$. Takie podejście prowadzi do MPA Moore'a U_4 (rys. 1). W MPA U_4 matryca MT_1^1 implementuje układ termów F^1, który zależy od zmiennych $T_r \in T$. Źródłem kodów klas $B_i \in \Pi_{RG}$ jest rejestr RG.



Rys. 1. Matrycowy układ automatu Moore'a U_4
Fig. 1. Matrix implementation of FSM U_4

Złożoność każdej z matryc można określić jako powierzchnię $S(MT_i)$ kryształu potrzebną dla jej realizacji. Złożoność układu logicznego MPA Moore'a U_4 może być wyrażona jako:

$$S(MT_1^1) = 2(L_1 + R)H_0^1; S(MT_1^2) = 2(L_2 + R_{CT})H_0^2;$$
$$S(MT_2) = H_0 R;\ S(MT_3) = 2RM;\ S(MT_4) = MN; \quad (3)$$
$$S(MT_5) = 2H(F^3)R;\ S(MT_6) = H(F^3)R_{CT}.$$

Porównajmy proponowaną metodę z MPA U_2 i U_3. Użyjemy probabilistycznego podejścia zaproponowanego w [5] i opracowanego w [3]. 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 operatorowych 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 operatorowym (warunk.).

2. Zastosowanie matrycowej implementacji układu logicznego jednostki sterującej [2].

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

Skorzystajmy z predstawionych w prace [3] oszacowań głównych parametrów MPA przedstawiono jako funkcje charakterystyk GSA i współczynników:

$$H_0 = 4.44 + 1.44 p_1 Q/p_3;\ H = 12.6 + 2.16 p_1 Q/p_3;$$
$$M = p_1 Q + 1;\ L = (1 - p_1)Q/p_4. \quad (4)$$

Q_0 jest liczbą zbiorów warunków wyjściowych, $p_3 = p_1 Q/Q_0 \in \{1.1;1.2\}$, $p_4 = p_2 Q/L \in \{1.1;1.2\}$.

Przeanalizujmy metodę optymalnego kodowania stanów oraz metodę transformacji kodów stanów. Zużycie zasobów sprzętowych dla MPA U_2 i U_3 jest zdefiniowane jako

$$S(U_2) = \sum_{i=1,4} S(M_i) = (p_5 H_0 + (1 - p_5)H_1)(3R + 2L) + M(2R + N); \quad (5)$$

$$S(U_3) = \sum_{i=1,6} S(M_i) = H_0(2(R_B + L) + R) + M(2R + N) + H(F^3)(2R + R_B). \quad (6)$$

$p_5 \in [0,1]$ jest współczynnikiem efektywności metody optymalnego kodowania. Przy $p_5 = 1$ wszystkie klasy stanów pseudoekwiwalentnych mogą być przedstawione jednym interwalem R-wymiarowej przestrzeni Boolowskiej Używając równania (5) i (6), możemy otrzymać odpowiednio $S(U_2) = f(p_1, Q, p_3, p_4, N, p_5)$ i $S(U_3) = f(p_1, Q, p_3, p_4, N)$. Równania końcowe są zbyt wielkie i mało informatywne, dlatego nie będziemy ich tu zamieszczać.

W podobny sposób możemy otrzymać $S(U_4) = f(p_1, Q, p_3, p_4, N, p_5, p_6)$, gdzie $H(F^3) = Mp_6$, $p_6 = \overline{0,5;0,8}$.

Analiza różnych GSA pokazała, że $Q \in \{100,\dots,1000\}$ i $N \le 100$. Przeanalizujmy funkcje $\eta_1 = S(U_4)/S(U_2)$ i $\eta_2 = S(U_4)/S(U_3)$. Z analizy wynika, że zużycie zasobów sprzętowych w modelu U_4 jest niższe niż w U_2 i w U_3. Najlepsze wyniki otrzymujemy dla GSA z liczbą wierzchołków $700 \le Q \le 1000$. Oprócz tego funkcje $\eta_{1,2}$ w istotny sposób zależą od p_5. Przy wzroście p_5 efektywność zaproponowanej metody w porównaniu z U_2 zmniejsza się, ale nawet przy $p_5 = 0,9$ można otrzymać zysk około 6 – 7 % w zależności od innych parametrów GSA. Maksymalny zysk przy $p_5 = 0,9$ wyniósł 7,5%. Odwrotna sytuacja jest przy porównaniu U_4 z U_3, przy $p_5 = 0,1$ można zyskać około 7 –8 % powierzchni. Maksymalny zysk przy $p_5 = 0,1$ wyniósł 8,1%.

4. Wnioski

Proponowana metoda dekompozycji matrycy MT_1 ma dwie zalety. Gwarantuje zmniejszenie liczby termów w układzie funkcji wzbudzeń pamięci MPA Moore'a do odpowiedniej wartości w ekwiwalentnym automacie Mealy'ego oraz wykorzystanie dwóch źródeł kodów stanów pozwala zmniejszyć złożoność układu bloku transformacji kodów stanów (w porównaniu do układu z jednym źródłem kodu stanów). Ponadto dekompozycja tabeli przejść na dwie pod-tabele umożliwia zmniejszenie złożoności bloku tworzenia termów układu funkcji wzbudzenia pamięci. Jest to związane z redukcją liczby wejściowych zmiennych określających termy dla każdej pod-tabeli w porównaniu ze znanymi metodami syntezy automatu Moore'a.

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.

5. Literatura

[1] Baranov S. Logic and system design of digital systems. Tallinn: TUT Press, 2008.
[2] Baranov S., Levin I., Koren O., Karpovski 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.
[3] Barkalov A., Titarenko L. Logic synthesis for FSM-based control units. Lecture notes in electrical engineering. Berlin: Springer Verlag Heidelberg, 2009, no. 53.
[4] Barkalov A., Titarenko L., Hebda O. Optimization of Moore finite-state-machine matrix circuit. Pomiary, Automatyka, Kontrola 2011; 57(8), 939-941.
[5] G. Novikov About one approach for finite state machines research, Contr. Syst. Mach. 1974; 2, 70–75, (in Russian).
[6] Hon A.D., Wilson M. Nanowire-based sublithographic programmable logic arrays. International Journal of Applied Mathematics and Computer Sciences 2007; 17(4), 565–575.
[7] 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.