

**Adam KLIMOWICZ**

POLITECHNIKA BIAŁOSTOCKA, ul. Wiejska 45a, 15-351 Białystok

## Zastosowanie metody syntezy wielopoziomowych układów kombinacyjnych do realizacji części kombinacyjnej wspólnego modelu automatów skończonych Mealy'ego i Moore'a

dr inż. Adam KLIMOWICZ

Ukończył studia w Instytucie Informatyki Politechniki Białostockiej. Obronił rozprawę doktorską w 2007 r. na Wydziale Informatyki Politechniki Białostockiej, gdzie zajmuje stanowisko asystenta. Jego zainteresowania naukowe to synteza układów kombinacyjnych i sekwencyjnych na bazie programowalnych układów logicznych, ze szczególnym uwzględnieniem układów o strukturze PLD/CPLD.

e-mail: a.klimowicz@pb.edu.pl



### Streszczenie

Praca dotyczy problemu syntezy układu sekwencyjnego w oparciu o programowalne układy logiczne. Cechą szczególną zastosowanej metody jest wykorzystanie wartości zmiennych wyjściowych jako kodu lub części kodu stanów wewnętrznych automatu skończonego. Do uproszczenia funkcji wyjść i funkcji wzbudzeń elementów pamięci automatu zastosowano metodę syntezy wielopoziomowych układów kombinacyjnych wykorzystującą sprzężenia zwrotne układu programowalnego.

**Słowa kluczowe:** automaty skończone, synteza logiczna, CPLD/FPGA

### An Application of Synthesis Method of Multilevel Combinational Circuits to Realization of Combinational Part of Combined Mealy and Moore Model of Finite State Machine

#### Abstract

In this paper, a problem of synthesis of sequential circuit on programmable logic device was presented. A special feature of the method is application of values of output variables as a code or as a part of code of internal states of a finite automata. The synthesis method of multilevel combinational circuits, which uses feedbacks of PLD macrocells to synthesis of finite state machines was applied to simplification of combinational part of state machine.

**Keywords:** finite state machines, logic synthesis, CPLD/FPGA

## 1. Wstęp

Problem wykorzystywania wartości zmiennych wyjściowych automatów skończonych jako części kodu stanu wewnętrznego od dawna budzi zainteresowanie projektantów systemów cyfrowych [1, 2, 3, 4]. Wraz z pojawieniem się programowalnych układów logicznych (PLD) w charakterze bazy technologicznej systemów cyfrowych, aktualnym stało się zadanie znalezienia efektywnych metod syntezy automatów skończonych na PLD. W tym celu zaproponowano sześć modeli strukturalnych automatów [4], które łatwo zrealizować w układach programowalnych oraz modele wspólne będące połączeniem modeli podstawowych.

Warunkiem połączenia w jeden model automatów różnych klas jest zgodność diagramów czasowych sygnałów wyjściowych dla wspólnych modeli. W pracach [5, 6] odstąpiono od warunku absolutnej zgodności tych diagramów dla modeli wspólnych. Osłabienie warunku pozwala na połączenie automatów klas A i C. W pracy proponuje się dodatkowo zastosować metodę syntezy wielopoziomowych układów kombinacyjnych [4, 7] do realizacji części kombinacyjnej automatu.

## 2. Modele strukturalne automatów skończonych

W praktyce najczęściej używa się dwóch typów automatów skończonych: automatu Mealy'ego [8] i automatu Moore'a [9].

Jeżeli każdy zbiór sygnałów wyjściowych  $w_i$  automatu Moore'a jest taki sam jak kod jego stanu wewnętrznego  $a_i$ , to zachowanie automatu można opisać w następujący sposób:

$$\begin{aligned} a_{t+1} &= \varphi(z_t, a_t); \\ w_t &= a_t. \end{aligned} \quad (3)$$

gdzie funkcja  $\varphi$  określa stan następny automatu  $a_{t+1}$  zależny od zbioru zmiennych wejściowych  $z_t$ , a formowany zbiór zmiennych wyjściowych  $w_t$  jest identyczny z kodem stanu.

Ten typ automatu został nazwany automatem klasy C [3, 4]. Jego cechą charakterystyczną jest to, że wszystkie jego wyjścia są wyjściami rejestrowymi. Można zauważyć, że dana struktura jest łatwa w realizacji, gdyż wszystkie współczesne układy PLD dopuszczają programowanie wyjść rejestrowych, a każde wyjście układu posiada sprzężenie zwrotne z częścią kombinacyjną PLD.

## 3. Wspólny model automatów skończonych klas A i C

Niech  $X = \{x_1, \dots, x_L\}$  – będzie zbiorem zmiennych wejściowych,  $Y = \{y_1, \dots, y_N\}$  – zbiorem zmiennych wyjściowych,  $A = \{a_1, \dots, a_M\}$  – zbiorem stanów wewnętrznych automatu skończonego

Przy projektowaniu automatów skończonych można spotkać sytuację, kiedy na wszystkich przejściach z pewnego stanu  $a_i$ ,  $a_i \in A$ , formowany jest ten sam wektor  $w_i$  wartości zmiennych wyjściowych. Jeżeli wektor wyjściowy  $w_i$  będzie kodem lub częścią kodu stanu  $a_i$ , to przy przejściu automatu do stanu  $a_i$ , w rejestrze zostanie utworzony kod stanu  $a_i$ , przy czym na wyjściach automatu zostanie uformowany wektor wyjściowy  $w_i$ . Pozwala to wykorzystywać wyjścia elementów pamięci w charakterze wyjść automatu skończonego. Do tego celu wystarczy makrokomórki wyjściowe PLD, które realizują elementy pamięci skonfigurować w trybie z wyjściami rejestrowymi.

Ponieważ wektor wyjściowy  $w_i$  jest formowany na wszystkich przejściach ze stanu  $a_i$ , to nie on zależy od zmiennych wejściowych, a tylko od stanu wewnętrznego  $a_i$ . W taki sposób wektor  $w_i$  jest wektorem wyjściowym automatu Moore'a. Oprócz tego, ponieważ kod (lub część kodu) stanu  $a_i$  jest zgodna z wektorem wyjściowym  $w_i$ , to wektor ten jest wektorem wyjściowym automatu klasy C. W ten sposób można utworzyć wspólny model automatu klasy AC.

Wykorzystanie wektora wyjściowego  $w_i$  jako kodu (lub części kodu) stanu  $a_i$  wymaga spełnienia poniższych warunków:

- 1) wektor wyjściowy  $w_i$  nie może być formowany na innych przejściach automatu, oprócz przejść ze stanu  $a_i$ ;
- 2) wszystkie zmienne przyjmujące wartość '1' w wektorze  $w_i$  powinny być zmiennymi wyjściowymi automatu Moore'a.

Idea podejścia polega na znalezieniu takich wektorów wartości zmiennych wyjściowych, które są formowane na wszystkich przejściach z odpowiednich stanów i spełniają powyższe warunki oraz na specjalnym kodowaniu stanów wewnętrznych automatu skończonego polegającym na wykorzystaniu wcześniej znalezionych wektorów wyjściowych jako kodów lub części kodów stanów wewnętrznych automatu skończonego [6].

Dodatkowo, w celu bardziej efektywnej realizacji części kombinacyjnej automatu, do jej budowy zastosowano metodę syntezy wielopoziomowych układów kombinacyjnych [7].

#### 4. Synteza wielopoziomowych układów kombinacyjnych

Metoda syntezy wielopoziomowych układów kombinacyjnych [4, 7] pozwala na realizację złożonych funkcji logicznych przy pomocy sprzężeń zwrotnych oraz metod faktoryzacji. Założenia rozpatrywanej metody syntezy są następujące:

- wartości już realizowanych funkcji i ich negacji stosuje się w charakterze podfunkcji do syntezy innych funkcji;
- wprowadza się funkcje pośrednie do wykorzystania w charakterze podfunkcji nie realizowanych dotąd funkcji;
- realizuje się tylko jedną z funkcji  $y_i$  lub  $\bar{y}_i$ , a ostatecznie potrzebną postać funkcji na wyjściu PLD tworzy się drogą zmiany logicznego poziomu sygnału wyjściowego.

Dla każdej funkcji  $y_i$  określa się wielkości  $Q(y_i)$  i  $Q(\bar{y}_i)$  - liczbę elementarnych koniunkcji oraz liczbę termów podłączonych do jednej makrokomórki koniecznych do realizacji funkcji,  $Q(\tilde{y}_i) = \min(Q(y_i), Q(\bar{y}_i))$ .

Układ programowalny jest opisany za pomocą zbioru termów podłączonych do poszczególnych makrokomórek  $Q = \{q_1, \dots, q_k\}$ , gdzie  $q_j$  - liczba termów podłączonych do makrokomórki  $MC_j$ ,  $q_j \in Q$ . Przy naruszeniu warunku :

$$Q(\tilde{y}_i) \leq q_j. \quad (5)$$

zadanie syntezy może być rozwiązane poprzez zbudowanie układu wielopoziomowego, w którym złożone funkcje otrzymujemy poprzez użycie wewnętrznych pętli sprzężenia zwrotnego PLD.

Niech  $Y^* = \{y_1, \dots, y_N, \bar{y}_1, \dots, \bar{y}_N\}$  będzie rozszerzonym zbiorem funkcji boolowskich;  $K(\tilde{y}_i)$  - zbiorem elementarnych koniunkcji  $\tilde{y}_i$ , gdzie  $\tilde{y}_i \in \{y_i, \bar{y}_i\}$ ,  $i = \overline{1, N}$ . Można zauważyć, że funkcja  $\tilde{y}_i$  jest podfunkcją dla funkcji  $\tilde{y}_j$  jeśli jest spełnione, że:

$$K(\tilde{y}_i) \setminus K(\tilde{y}_j) = \emptyset. \quad (6)$$

Efektywność wykorzystania funkcji  $\tilde{y}_i$ ,  $\tilde{y}_j \in Y^*$ , w charakterze podfunkcji funkcji  $\tilde{y}_j$ ,  $\tilde{y}_j \in Y^*(\tilde{y}_j)$ , można wyrazić w postaci:

$$d(\tilde{y}_i, \tilde{y}_j) = |K(\tilde{y}_j)| - (|K(\tilde{y}_i) \setminus K(\tilde{y}_j)| + 1). \quad (7)$$

Wówczas ogólną efektywność funkcji  $\tilde{y}_i$ , w charakterze podfunkcji całego zbioru  $Y^*$ , można przedstawić w postaci:

$$D(\tilde{y}_i) = \sum_{\tilde{y}_j \in Y^*(\tilde{y}_i)} d(\tilde{y}_i, \tilde{y}_j). \quad (8)$$

Przez  $Q_i$  zostanie oznaczona ranga funkcji  $y_i$  (liczba termów potrzebnych do realizacji)  $Q_i = \min(|K(y_i)|, |K(\bar{y}_i)|)$ . Zadanie syntezy można uznać za rozwiązane, jeśli przy dostatecznej liczbie makrokomórek do realizacji wszystkich pośrednich funkcji, dla każdej funkcji  $y_i$ , znajdzie się dowolna makrokomórka  $MC_j$  taka, że spełnione są warunki:

$$Q_i \leq q_j, q_j \in Q \quad (9)$$

przy czym w pierwszej kolejności realizujemy funkcje o największej efektywności.

#### 5. Badania eksperymentalne

Badania eksperymentalne przeprowadzono na układach testowych z MCNC [10]. Do badań użyto po jednym układzie z rodzin CPLD MAX3000A oraz FPGA FLEX10KE firmy Altera i systemu MAX+Plus II.

Rezultaty badań pokazują, że przy syntezie automatów dodatkowa metoda syntezy części kombinacyjnej pozwoliła zredukować w części przypadków koszt układu w stosunku do metody AC (średnio o 6%) przy jednoczesnym zmniejszeniu szybkości układu (średnio o 7%). Jednocześnie koszt realizacji był zawsze niższy niż dla metody klasycznej.

#### 6. Wnioski

Zastosowanie metody syntezy układów wielopoziomowych do syntezy części kombinacyjnej automatu przy realizacji wspólnego modelu automatów klas A i C pozwoliło w większości przypadków zmniejszyć koszt realizacji wyrażony liczbą wykorzystanych komórek logicznych oraz w kilku przypadkach zredukować opóźnienie zarówno w stosunku do metody klasycznej, jak i metody syntezy modelu AC bez dodatkowej syntezy części kombinacyjnej. Szczególnie dobre rezultaty otrzymano dla układu FPGA, co związane jest z tym, że realizowane funkcje logiczne były zależne od dość małej liczby zmiennych wejściowych, co jest ważne w przypadku realizacji funkcji na układach typu LUT (Look-Up Table) mających małą liczbę wejść (np. 4).

#### 7. Literatura

- [1] Pomeranz I., Cheng K.-T. STOIC: state assignment based on output/input functions // IEEE Transactions on CAD, August 1993. - Vol. 12, 8, pp. 613-622.
- [2] Forrest J.: ODE: output direct state machine encoding // Proc. of the European Design Automation Conference (EURO-DAC'95), 1995, pp.600-605.
- [3] Solovjev V. Synthesis of Sequential Circuits on Programmable Logic Devices Based on New Models of Finite State Machines // Proc. of the Euromicro Symposium on Digital System Design (DSD'2001), September 4-6, 2001, Warsaw, Poland, pp. 170-173.
- [4] Salauyou V., Klimowicz A. Synteza logiczna układów cyfrowych w strukturach programowalnych // Oficyna Wydawnicza Politechniki Białostockiej, Białystok 2010.
- [5] Klimowicz A., Salauyou V. The Synthesis of Combined Mealy and Moore Machines Structural Model Using Values of Output Variables as Codes of States // Proc. Of 15th Euromicro Conference on Digital System Design (DSD 2012), Izmir, September 5-8, 2012, pp. 789-794
- [6] Klimowicz A., Salauyou V. Zastosowanie wspólnego modelu automatów Mealy'ego i Moore'a do realizacji układów sekwencyjnych w układach programowalnych // Pomiar. Automatyka. Kontrola. - Vol. 58, nr 7 (2012), s. 653-655
- [7] Klimowicz A., Sołowjew W.: Automatykacja projektowania kontrolerów typu kombinacyjnego na PLD z wykorzystaniem sprzężeń zwrotnych // Materiały XIV Krajowej Konferencji Automatyki (Zielona Góra 24-27 czerwca 2002), Tom II. - s. 751-756.
- [8] Mealy G.H. A method for synthesizing sequential circuits // Bell System Techn. J., Vol.34, 1955, pp. 1045-1079.
- [9] Moore E.F. Gedanken-experiments on sequential machines // In C. Shannon and J.McCarthy editors // Automata Studies Princeton University Press, 1956, pp. 129-153.
- [10] Yang S.: Logic Synthesis and Optimization Benchmarks User Guide; Microelectronics Center of North Carolina, Research Triangle Park, NC, 1991.

Artykuł wykonano w ramach pracy statutowej S/WI/1/2013 Politechniki Białostockiej.