

Adam KLIMOWICZ, Valery SALAUYOU

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

Zastosowanie wspólnego modelu automatów Mealy'ego i Moore'a do realizacji układów sekwencyjnych w układach programowalnych

Dr inż. Adam KLIMOWICZ



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

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

Dr hab. inż. Valery SALAUYOU



Ukończył studia na Wydziale Matematyki Stosowanej w Białoruskim Państwowym Uniwersytecie w Mińsku w 1978 r. W 1986 r. obronił rozprawę doktorską, a w 2003 r. uzyskał tytuł doktora habilitowanego. Pracuje na Wydziale Informatyki Politechniki Białostockiej, gdzie zajmuje stanowisko adiunkta. Od 25 lat pracuje w dziedzinie projektowania logicznego systemów cyfrowych.

e-mail: valsol@mail.ru

Streszczenie

W pracy opisano syntezę automatów skończonych na bazie programowalnych układów logicznych (PLD). Cechą szczególną metody jest zastosowanie wartości zmiennych wyjściowych w charakterze części kodu stanów wewnętrznych automatu. W celu rozwiązania zadania został zastosowany wspólny model automatów Mealy'ego i Moore'a, przy czym automat nie podlega żadnym przekształceniom związanym ze zwiększeniem liczby stanów wewnętrznych i liczby przejść. W pracy opisano też metodę syntezy wspólnego modelu automatów skończonych klas AC.

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

Application of the common model of Mealy and Moore finite state machines to realization of sequential circuits on programmable logic devices

Abstract

This paper describes the problem of synthesis of finite automata on programmable logic devices. A special feature of the method is the application of the values of output variables as a code or the part of a code of internal states of finite automata. In order to solve the problem, a common model of Mealy [4] and Moore [5] machines is used. The main difference of this approach in relation to known methods [1-3, 6, 7, 9] is that the finite state machine does not undergo any transformation associated with an increase in the number of internal states and the number of transitions of a finite automaton. In this paper three models of finite state machines are considered (classes: A, B and C). They are applied to realization of a FSM on programmable logic. The paper presents the necessary conditions for the possibility of using the values of output variables as a code of internal states of a finite automaton. In the paper there is described the method for synthesis of a common model for the finite state machine of AC class. The idea of the proposed approach is to find such sets of the values of output variables which are formed at all transitions from the corresponding states and satisfy the conditions of realization. It also aims at doing a special coding of the internal states, where the sets of values of the output variables are used as a part of the code of the internal states. There are given possible directions for future research in the area of synthesis of new structural models of finite state machines.

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

1. Wstęp

Problem wykorzystywania wartości zmiennych wejściowych i wyjściowych automatów skończonych w charakterze części kodu stanu wewnętrznego od dawna cieszy się zainteresowaniem projektantów systemów cyfrowych [1, 2, 3, 6, 7, 9]. Podobne podejścia w wielu przypadkach pozwoliło znacznie obniżyć koszt realizacji i zwiększyć szybkość działania automatu skończonego.

Wraz z pojawieniem się programowalnych układów logicznych (PLD) [8], takich jak Complex Programmable Logic Devices (CPLD) i Field Programmable Gate Arrays (FPGA), 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 [6], które zostały nazwane automatami klas A, B, C, D, E i F. Ich zaletą jest łatwość realizacji w układach programowalnych. W pracy [6] przeprowadzono analizę czasową modeli dla przypadku maksymalnej szybkości działania automatu skończonego, określono minimalny czas trwania taktu automatu (minimalny okres synchronizacji) i maksymalną częstotliwość funkcjonowania automatu skończonego. W pracy [6] zaproponowano także cztery wspólne modele automatów skończonych: ADE, AD, AE i BF.

Głównym warunkiem możliwości połączenia w jednym modelu automatów różnych klas jest zgodność diagramów czasowych sygnałów wyjściowych dla wspólnych modeli. Ostatni warunek znacznie ogranicza możliwość połączenia różnych modeli. W praktyce często spotyka się automaty skończone, które częściowo mają cechy różnych modeli. Jednak z powodu podanego wyżej ograniczenia zwykle wykorzystuje się najbardziej ogólne modele automatów: Mealy'ego [4] i Moore'a [5].

W tej pracy przy realizacji automatów skończonych na PLD proponuje się odstąpić od warunku absolutnej zgodności diagramów czasowych sygnałów wyjściowych dla modeli wspólnych. Wystarczy, aby sygnały wyjściowe we wspólnych modelach były formowane w przedziałach tego samego taktu automatu. Takie osłabienie warunku możliwości połączenia modeli, w porównaniu z pracą [6], pozwoli na połączenie automatów skończonych klas A i C, co w niektórych przypadkach może prowadzić do znacznej redukcji kosztu realizacji bez obniżania szybkości działania automatu skończonego.

2. Podstawowe modele strukturalne automatów skończonych

W praktyce najbardziej rozpowszechnione są dwa typy automatów skończonych: automat Mealy'ego i automat Moore'a. Zachowanie automatu Mealy'ego opisuje się za pomocą następujących równań:

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

gdzie funkcja φ określa stan następny automatu a_{t+1} , a funkcja wyjść ψ - formowany zbiór zmiennych wyjściowych w_t . Cechą szczególną automatu Mealy'ego jest to, że funkcja wyjść zależy od zbioru zmiennych wejściowych z_t oraz od stanu wewnętrznego a_t .

W automacie Moore'a funkcja wyjść zależy tylko od stanu obecnego a_t , nie zależy natomiast od zmiennych wejściowych. Dlatego działanie automatu Moore'a opisuje się w sposób następujący:

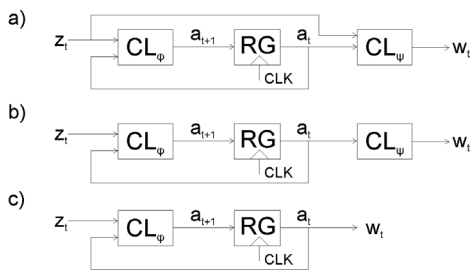
$$\begin{aligned} a_{t+1} &= \varphi(z_t, a_t), \\ w_t &= \psi(a_t). \end{aligned} \quad (2)$$

Funkcje przejść φ i wyjść ψ automatu realizowane są za pomocą układów kombinacyjnych CL_φ i CL_ψ , a pamięć automatu jest realizowana w postaci rejestru RG, gdzie przechowywany jest kod stanu wewnętrznego a_t . Struktura automatów Mealy'ego i Moore'a jest pokazana, odpowiednio, na rys. 1-a i 1-b.

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

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

Ten typ automatu został nazwany automatem klasy C [6]. Struktura automatu klasy C jest pokazana na rys. 1-c. Jej charakterystyczną cechą jest brak układu kombinacyjnego CL_ψ , a także to, że wszystkie wyjścia automatu są wyjściami rejestrowymi, ponieważ są one wyjściami rejestru RG. Należy zauważyć, że dana struktura jest prosta do realizacji na PLD, ponieważ wszystkie współczesne układy PLD dopuszczają programowanie wyjść rejestrowych, a każde wyjście posiada sprzężenie zwrotne z częścią kombinacyjną układu PLD.



Rys. 1. Podstawowe struktury automatów:
a – klasy A, b – klasy B, c – klasy C
Fig. 1. Basic structures of FSMs:
a – class A, b – class B, c – class C

3. Proponowane podejście

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 wewnętrznego automatu skończonego.

Przy projektowaniu automatów skończonych często zdarza się sytuacja, kiedy na wszystkich przejściach z pewnego stanu a_i , $a_i \in A$, formowany jest ten sam wektor w_i ; wartości zmiennych wyjściowych. Niech wektor wyjściowy w_i będzie kodem lub częścią kodu stanu a_i . Wówczas przy przejściu automatu do stanu a_i , w rejestrze RG zostanie utworzony kod stanu a_i , przy czym na wyjściach automatu zostanie uformowany wektor wyjściowy w_i . Pozwala to wykorzystywać wyjścia elementów pamięci, które odpowiadają wartościom równym '1' wektora w_i , w charakterze wyjść automatu skończonego. Do tego celu wystarczy makrokomórki wyjściowe PLD, które realizują dane elementy pamięci skonfigurować w trybie wyjść rejestrowych. W taki sposób, na tych samych makrokomórkach będą realizowane zmienne wyjściowe i zmienne pętli sprzężenia zwrotnego.

Ponieważ wektor wyjściowy w_i jest formowany na wszystkich przejściach ze stanu a_i , to wektor wyjściowy w_i nie zależy od zmiennych wejściowych, a tylko od stanu wewnętrznego a_i . W taki sposób wektor wyjściowy w_i faktycznie 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 wyjściowy w_i jest wektorem wyjściowym automatu klasy C. W taki sposób powstał wspólny model automatu klasy AC.

Twierdzenie 1.

Wykorzystanie wektora wyjściowego w_i w charakterze kodu (lub części kodu) stanu a_i wymaga spełnienia następujących warunków koniecznych:

- 1) wektor wyjściowy w_i nie powinien być formowany na innych przejściach automatu skończonego, oprócz przejść ze stanu a_i ;
- 2) wszystkie zmienne wyjściowe, przyjmujące wartość '1' w wektorze w_i powinny być zmiennymi wyjściowymi automatu Moore'a, tzn. nie powinny zależeć od zmiennych wejściowych.

Dowód

Naruszenie warunku (1) może doprowadzić do nieprzewidzianego przejścia automatu do stanu a_i , czyli naruszenia warunku determinizmu funkcjonowania automatu skończonego. Naruszenie warunku (2) czyni niemożliwą zmianę wartości zmiennych wyjściowych automatu Mealy'ego przy zmianie wartości zmiennych wejściowych, czyli prowadzi do naruszenia algorytmu funkcjonowania automatu skończonego.

Idea proponowanego podejścia polega na znalezieniu takich wektorów wartości zmiennych wyjściowych, które są formowane na wszystkich przejściach z odpowiednich stanów i spełniają warunki określone w twierdzeniu 1 oraz na specjalnym kodowaniu stanów wewnętrznego automatu skończonego, przy którym wcześniej znalezione wektory wyjściowe są wykorzystywane jako kody lub części kodów stanów wewnętrznego.

Można zauważyć pewne różnice pomiędzy proponowanym podejściem, a analogicznymi znanymi metodami syntezy automatów skończonych. W pracy [6] połączenie automatów klas A i C uważa się za niemożliwe, ponieważ dla automatów klas A i C nie zachodzi zgodność diagramów czasowych formowanych sygnałów wyjściowych. W proponowanym podejściu wykorzystywany jest wspólny model automatów klas A i C.

Niech na początku taktu automatu nastąpi przejście automatu w stabilny stan wewnętrzny, tzn. kiedy na wyjściach elementów pamięci będą uformowane właściwe wartości. Przy tradycyjnej realizacji automatów Mealy'ego na PLD od początku taktu sygnały wyjściowe uformują się nie wcześniej niż po czasie opóźnienia t_{PD} wprowadzanego przez część kombinacyjną układu PLD. W przedstawionym podejściu sygnały wyjściowe automatu Moore'a określone przy pomocy kodu stanu wewnętrznego automatu będą uformowane na samym początku taktu automatu czyli szybciej o czas t_{PD} , niż sygnały wyjściowe automatu Mealy'ego. W większości przypadków praktycznego zastosowania automatów skończonych takie formowanie sygnałów wyjściowych jest dopuszczalne.

W metodzie syntezy automatu klasy C opisanej w [6] obowiązkowe jest przejście od automatu Mealy'ego do automatu Moore'a, co zwykle jest związane ze zwiększeniem liczby stanów wewnętrznych i liczby przejść automatu skończonego. W proponowanym podejściu dany automat skończony nie podlega żadnym przekształceniom związanym ze zwiększeniem liczby stanów wewnętrznych i liczby przejść automatu skończonego.

4. Metoda syntezy wspólnego modelu automatów skończonych klas A i C

Niech A – zbiór stanów wewnętrznych pewnego automatu skończonego. Stanem wewnętrznym automatu Moore'a będzie nazywany taki stan automatu $a_i \in A$, na którego wszystkich przejściach jest formowany ten sam wektor zmiennych wyjściowych.

Niech zbiór A_B zawiera wszystkie stany automatu Moore'a dla danego automatu oraz niech $A_B \subseteq A$. Pozostałe stany wewnętrzne automatu tworzą zbiór A_A stanów automatu Mealy'ego, $A_A = A \setminus A_B$.

Niech $y(a_i)$ będzie zbiorem zmiennych wyjściowych, których wartości równe '1' są formowane na przejściach ze stanu $a_i \in A$. Wyjściowymi zmiennymi automatu Mealy'ego zbioru Y_A będą określone takie zmienne wyjściowe automatu skończonego, których wartości równe '1' są formowane na przejściach ze stanów automatu Mealy'ego – ze zbioru A_A :

$$Y_A = \{ y(a_i) \mid a_i \in A_A \}. \quad (4)$$

Pozostałe zmienne wyjściowe automatu skończonego ze zbioru Y_B będą określone zmiennymi wyjściowymi automatu Moore'a, $Y_B = Y \setminus Y_A$.

Objaśnijmy teraz szczegóły realizacji na PLD zmiennych wyjściowych automatów Mealy'ego i Moore'a. Zmienne wyjściowe automatu Mealy'ego ze zbioru Y_A bezpośrednio zależą od zmiennych wejściowych i są asynchroniczne w stosunku do zmiennych wejściowych, tzn. przy zmianie wartości odpowiednich zmiennych wejściowych mogą zmieniać się wartości zmiennych wyjściowych automatu Mealy'ego. Dlatego zmienne wyjściowe automatu Mealy'ego ze zbioru Y_A zawsze będą realizowane na wyjściach kombinacyjnych układu PLD.

Zmienne wyjściowe automatu Moore'a ze zbioru Y_B nie zależą bezpośrednio od zmiennych wejściowych, ich wartości zmieniają się razem ze zmianą stanu wewnętrznego automatu skończonego. Wartości zmiennych wyjściowych automatu Moore'a mogą występować w charakterze kodu lub części kodu stanów wewnętrznych. Do realizacji zmiennych wyjściowych automatu Moore'a wykorzystuje się konfigurację makrokomórek PLD z wyjściami rejestrowymi, przy czym sprzężenia zwrotne makrokomórek są wykorzystywane do przekazywania wartości zmiennych sprzężeń zwrotnych na wejście części kombinacyjnej automatu skończonego.

Z uwzględnieniem powyższych założeń algorytm syntezy wspólnego modelu automatów skończonych klas A i C jest następujący:

Algorytm 1.

1. Określa się zbiór Y_B zmiennych wyjściowych automatu Moore'a. Jeżeli $Y_B = \emptyset$, to należy wykorzystać jedną z metod syntezy automatu Mealy'ego [6], wykonuje się przejście do punktu 6.
2. Buduje się macierz W , której elementy mogą przyjmować trzy wartości: '0', '1' i '-' (wartość nieokreślona). Wiersze macierzy W odpowiadają stanom wewnętrznym automatu skończonego, a kolumny – zmiennym wyjściowym automatu Moore'a ze zbioru Y_B . Dla stanów wewnętrznych automatu Moore'a ze zbioru A_B w odpowiednich wierszach macierzy W zapisuje się wartości zmiennych wyjściowych automatu Moore'a ze zbioru Y_B , formowane na przejściach z odpowiednich stanów. Dla stanów wewnętrznych automatu Mealy'ego ze zbioru A_A w odpowiednich wierszach macierzy W wstawia się wartości nieokreślone ('-').
3. Rozwiązuje się zadanie dołączenia do wierszy macierzy W kodu binarnego o minimalnej długości R , w taki sposób, aby wszystkie wiersze macierzy W były wzajemnie ortogonalne.
4. Określa się kody stanów wewnętrznych automatu. W tym celu dla każdego stanu $a_i \in A$, ustala się kod $K(a_i)$ równy wartościom i -tego wiersza macierzy W .

5. Buduje się równania logiczne funkcji przejść i funkcji wyjść automatu skończonego.

6. Koniec.

Do rozwiązania zadania dołączenia do macierzy W kodu binarnego o minimalnej długości (punkt 3) można zastosować metodę opisaną w pracy [6] w rozdziale 4. Idea metody polega na zbudowaniu grafu ortogonalności wierszy macierzy W i znalezienie w grafie minimalnej liczby rozdzielných pełnych podgrafów (klik) i ich zakodowaniu kodem binarnym o minimalnej długości.

5. Wnioski

Odstępianie od warunku absolutnej zgodności diagramów czasowych sygnałów wyjściowych dla wspólnych modeli automatów skończonych pozwala znacznie rozszerzyć zbiór wspólnych modeli dopuszczających ich efektywną realizację w układach programowalnych. Dlatego perspektywnym kierunkiem dalszych badań wydaje się być opracowanie nowych wspólnych modeli strukturalnych automatów skończonych i ich metod syntezy w strukturach programowalnych, a także opracowanie nowych modeli strukturalnych automatów skończonych, które pozwalają efektywnie wykorzystywać właściwości współczesnych układów programowalnych o złożonej architekturze.

Artykuł powstał w ramach pracy statutowej nr S/WI/4/2008 Politechniki Białostockiej.

6. Literatura

- [1] Pomeranz I., Cheng K.T.: State assignment using input/output functions. Proc. 29th ACM/IEEE Design Automation Conference (DAC), 1992, p. 573-577.
- [2] 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.
- [3] Forrest J.: ODE: output direct state machine encoding. Proc. of the European Design Automation Conference (EURO-DAC'95), 1995, pp.600-605.
- [4] Mealy G.H.: A method for synthesizing sequential circuits. Bell System Techn. J., Vol.34, 1955, pp. 1045-1079.
- [5] Moore E.F.: Gedanken-experiments on sequential machines. In C. Shannon and J.McCarthy (editors), Automata Studies, Princeton University Press, 1956, pp. 129-153.
- [6] Salauyou V., Klimowicz A.: Synteza logiczna układów cyfrowych w strukturach programowalnych. Oficyna Wydawnicza Politechniki Białostockiej, Białystok 2010.
- [7] Soloviev V.V.: Sintez mikroprogramnych avtomatov na progrmirujemych matricach logiki. Wiesci Akademii Nauk Belarusi. Ser. fiz.-techn. nauk, 1994, Nr 1. - s. 68-72.
- [8] Solovjev V.V., Vasiljev A.G.: Programmirujemyje logiczeskije integralne schiemy i ich primienienije. Mińsk: Bielaruskaja Nauka, 1998.
- [9] 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 Systems Design (DSD'2001), September 4-6, 2001, Warsaw, Poland, pp. 170-173.

otrzymano / received: 02.04.2012

przyjęto do druku / accepted: 01.06.2012

artykuł recenzowany / revised paper