Konferencja naukowa KNWS'04
„Informatyka - sztuka czy rzemiosło"
23-25 czerwca 2004, Czocha

SYNTEZA AUTOMATÓW MEALY'EGO Z WIELOKROTNYMI KODAMI STANÓW WEWNĘTRZNYCH

Arkadiusz Bukowiec

Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50

e-mail: a.bukowiec@iie.uz.zgora.pl

STRESZCZENIE

W referacie zostanie przedstawiona metoda sprzętowej optymalizacji automatu Mealy'ego. Bazuje ona na przypisaniu różnych kodów stanom wewnętrznym dla każdej podtablicy tabeli przejść-wyjść automatu. Przykład ilustrujący zastosowanie metody również zostanie zaprezentowany.

1. WPROWADZENIE

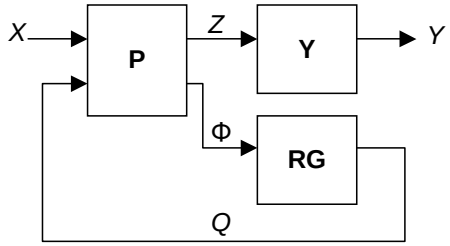
Skończone automaty stanów (ang. *Finite State Machine*, FSM) są wciąż jedną z najpopularniejszych metod realizacji układów logicznych. Dają one możliwość realizacji takiego układu w strukturach programowalnych (ang. *Programmable Logic Device*, PLD) takich jak CPLD lub FPGA. Wysoki koszt realizacji takiego układu powoduje, że optymalizacja wykorzystywanych zasobów sprzętowych stanowi wciąż ważny problem. Jednym z rozwiązań tego problemu jest zwiększenie liczby poziomów automatu [2]. Podejście takie wymaga utworzenia dodatkowych zmiennych wewnętrznych oraz bardzo często wykorzystania dodatkowych zasobów sprzętowych. Metoda opierająca się na wielokrotnym kodowaniu stanów wewnętrznych przedstawiona w tym referacie pozwala na zmniejszenie wykorzystywanych zasobów sprzętowych układu logicznego.

2. PODSTAWOWE DEFINICJE I IDEA METODY

Niech automat skończony z wyjściami typu Mealy'ego zostanie opisany za pomocą tablicy przejść-wyjść automatu [1, 4] z kolumnami: a_m będącą początkowym stanem automatu, $a_m \in A$ gdzie $A = \{a_1, \ldots, a_M\}$ jest zbiorem stanów automatu; $K(a_m)$ będącą binarnym kodem stanu $a_m \in A$, zakodowanym na $R =]\log_2 M[$ bitach, reprezentowanych przez zmienną $Q_r \in Q = \{Q_1, \ldots, Q_R\}$; a_s będącą stanem przejścia; $K(a_s)$ będącą kodem stanu $a_s \in A$; X_h będącą sygnałem wejściowym wymuszającym przejście $\langle a_m , a_s \rangle$ i równym koniunkcji pewnych elementów ze zbioru warunków logicznych $X = \{x_1, \ldots, x_L\}$; Y_h będącą sygnałem wyjściowym przypisanym do przejścia $\langle a_m , a_s \rangle$, $Y_h \subseteq Y$ gdzie $Y = \{y_1, \ldots, y_N\}$ jest zbiorem

mikrooperacji; Φ_h będącą podzbiorem zbioru funkcji wzbudzających Φ, których wzbudzenie powoduje przełączenie pamięć automatu z kodu $K(a_m)$ na kod $K(a_s)$, $\boldsymbol{\Phi} = \{\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_R\}$; h jest liczbą przejść automatu $h \in \{1, \ldots, H\}$.

Niech tablica przejść-wyjść automatu zawiera T różnych zbiorów mikrooperacji (mikroinstrukcji) $Y_t \subseteq Y$. Zakodujmy każdy taki zbiór $Y_t \subseteq Y$ binarnym kodem $K(Y_t)$ na $R_1 =]\log_2 T[$ bitach. Do reprezentacji tego kodu wykorzystamy zmienną $z_r \in Z = \{z_1, \ldots, z_{R1}\}$. W takim przypadku automat Mealy'ego może zostać zaimplementowany jako dwupoziomowy automat-PY (rys. 1).



Rys. 1. Schemat automatu-PY

W przypadku takim układ **P** realizuje system funkcji:

$$Z = Z(Q, X),$$

$$\Phi = \Phi(Q, X). \qquad (1)$$

W celu realizacji systemu (*1*) należy przekształcić początkową tablice przejść-wyjść automatu. Układ **Y** implementuje funkcje:

$$Y = Y(Z). \qquad (2)$$

Układ **RG** stanowi pamięć automatu i jest zbudowany z przerzutników typu D.

Takie podejście pozwala na zmniejszenie zasobów sprzętowych automatu Mealy'ego w porównaniu ze standardową realizacją jednopoziomową (automat-P) [3], gdzie nie występuje układ **Y** a układ **P** realizuje system funkcji:

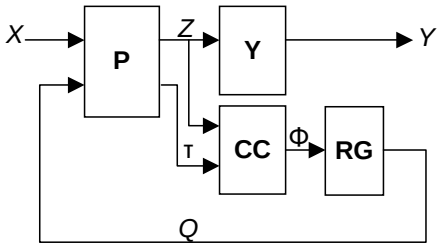
$$Y = Y(Q, X),$$

$$\Phi = \Phi(Q, X). \qquad (3)$$

Ideą metody zaprezentowanej w tym referacie jest zmniejszenie liczby wyjść układu **P** w porównaniu z automatem-PY. Niech $A(a_m) \subseteq A$ jest zbiorem stanów przejść ze stanu $a_m \in A$ i $|A(a_m)| = B_m$. Przez B_0 oznaczmy maksymalną liczbę różnych stanów przejść z jednego stanu a_m:

$$B_0 = \max(B_1, \ldots, B_M) \qquad (4)$$

i niech $R_3 =]\log_2 B_0[$. Zakodujmy każdy stan $a_s \in A(a_m)$ R_3-bitowym kodem $K_m(a_s)$. Niech kod ten będzie reprezentowany przez zmienne $\tau_r \in \tau = \{\tau_1, \ldots, \tau_{R3}\}$. W przypadku takim każdy stan $a_s \in A(a_m)$ może być zdeterminowany przez parę $\langle K(Y_q), K_m(a_s)\rangle$. Automat Mealy'ego z takim kodowaniem stanów wewnętrznych może zostać zrealizowany jako automat-PAY (rys. 2).



Rys. 2. Schemat automatu-PAY

W tym przypadku układ **P** realizuje następujący system funkcji:

$$Z = Z(Q, X),$$

$$\tau = \tau(Q, X). \tag{5}$$

Działanie układu **Y** jest zgodne z funkcją (*2*), natomiast wprowadzony zostaje dodatkowy układ konwersji kodu **CC** realizujący następującą funkcję:

$$\Phi = \Phi(Z, \tau). \tag{6}$$

Takie podejście pozwala na zmniejszenie wymaganej ilości wyjść układu **P** z $R + R_1$ (automat-PY) do $R_3 + R_1$. W pewnych warunkach pozwala to na zmniejszenie zasobów sprzętowych wymaganych do realizacji automatu Mealy'ego.

3. OPIS METODY ORAZ PRZYKŁAD JEJ ZASTOSOWANIA

W rozdziale tym przedstawiony jest sposób syntezy automatu Mealy'ego z wykorzystaniem proponowanej struktury automatu-PAY. Metoda ta jest zilustrowana przykładem przedstawionym za pomocą tablicy przejść-wyjść automatu (tab. 1).

Metoda składa się z następujących 6 kroków:

1. Kodowanie zbiorów mikrooperacji. W naszym przykładzie wyróżniamy $T = 7$ rożnych zbiorów mikrooperacji (tab. 1): $Y_1 = \{y_1, y_2\}$, $Y_2 = \{y_3\}$, $Y_3 = \{y_2, y_4\}$, $Y_4 = \{y_3, y_4\}$, $Y_5 = \{y_5\}$, $Y_6 = \{y_3, y_5\}$, $Y_7 = \{y_2\}$. Dlatego $R_1 = 3$, $Z = \{z_1, z_2, z_3\}$. Przyjmijmy, zatem $K(Y_1) = 000$, …, $K(Y_7) = 110$.

Tab. 1. Tablica przejść-wyjść automatu S_1

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Y_h	Φ_h	h
a_1	000	a_2	001	x_1	y_1y_2	D_3	1
		a_3	010	$\neg x_1\, x_2$	y_3	D_2	2
		a_2	001	$\neg x_1\, \neg x_2$	y_2y_4	D_3	3
a_2	001	a_3	010	x_2	y_3y_4	D_2	4
		a_4	011	$\neg x_2$	y_5	D_2D_3	5
a_3	010	a_5	100	x_3	y_1y_2	D_1	6
		a_4	011	$\neg x_3$	y_3y_5	D_2D_3	7
a_4	011	a_1	000	x_1	y_2	-	8
		a_5	100	$\neg x_1$	y_1y_2	D_1	9
a_5	100	a_4	011	1	y_3y_4	D_2D_3	10

2. Obliczenie długości kodu dla stanów wewnętrznych. Z tab. 1. mamy $A(a_1) = \{a_2, a_3\}$, $A(a_2) = \{a_3, a_4\}$, $A(a_3) = \{a_5, a_4\}$, $A(a_4) = \{a_1, a_5\}$, $A(a_5) = \{a_4\}$. Więc $B_1 = B_2 = B_3 = B_4 = 2$, $B_5 = 1$ z równania (*4*) wynika, że $B_0 = 2$, $R_3 = 1$, $\tau = \{\tau_1\}$.

3. Wielokrotne kodowanie stanów wewnętrznych. Zakodujmy stany $a_s \in A(a_m)$ za pomocą R_3-bitowego kodu. W celu tym utworzona zostaje tabela wielokrotnego kodowania stanów z kolumnami: $A(a_1)$, $K_1(a_s)$, …, $A(a_M)$, $K_m(a_s)$. Dla zaprezentowanego przykładu jest to tab. 2.

Tab. 2. wielokrotne kodowanie stanów automatu S_1

$A(a_1)$	$K_1(a_s)$	$A(a_2)$	$K_2(a_s)$	$A(a_3)$	$K_3(a_s)$	$A(a_4)$	$K_4(a_s)$	$A(a_5)$	$K_5(a_s)$
a_2	0	a_3	0	a_4	0	a_1	0	a_4	0
a_3	1	a_4	1	a_5	1	a_5	1	-	-

4. Utworzenie przekształconej tablicy przejść-wyjść automatu. W celu utworzenia systemu (*5*) należy przekształcić początkową tablicę przejść-wyjść automatu. Kolumna $K(a_s)$ zostaje zastąpiona kolumną $K_m(a_s)$ z kodami pobranymi z tabeli wielokrotnego kodowania stanów. Kolumna Y_h zostaje zastąpiona kolumną Z_h, która zawiera zmienne $z_r \in Z$ równe 1 w kodzie $K(Y_t)$ zbioru $Y_t \subseteq Y$ z h-tej linii początkowej tablicy przejść-wyjść automatu. Kolumna Φ_h zostaje zastąpiona kolumną τ_h, która zawiera zmienne $\tau_r \in \tau$ równe 1 w kodzie $K_m(a_s)$ odpowiedniego stanu przejścia z h-tego wiersza początkowej tablicy przejść-wyjść automatu.

Przekształcona tablica przejść-wyjść automatu dla naszego przykładu przedstawiona jest w tab. 3.

5. Utworzenie tablicy dla układu konwersji kodu. W celu realizacji systemu (*6*) należy utworzyć tablice dla układu konwersji kodu CC. Wiersze tej tablicy oznaczone są przez stany $a_m \in A$.

Tab 3. Przekształcona tablica przejść-wyjść automatu S_1

a_m	$K(a_m)$	a_s	$K_m(a_s)$	X_h	Z_h	τ_h	h

a_1	000	a_2	0	x_1	z_3	-	1
		a_3	1	$\neg x_1\, x_2$	z_2	τ_1	2
		a_2	0	$\neg x_1\, \neg x_2$	$z_2\, z_3$	-	3
a_2	001	a_3	0	x_2	z_1	-	4
		a_4	1	$\neg x_2$	$z_1\, z_3$	τ_1	5
a_3	010	a_5	1	x_3	z_3	τ_1	6
		a_4	0	$\neg x_3$	$z_1\, z_2$	-	7
a_4	011	a_1	0	x_1	z_2	-	8
		a_5	1	$\neg x_1$	z_3	τ_1	9
a_5	100	a_4	0	1	z_1	-	10

Zawiera ona następujące kolumny: a_s, $K_m(a_s)$, $K(Y_t)$, $K(a_s)$, Φ_s, h gdzie kolumna Φ_s zawiera funkcje wzbudzeń przerzutników **RG** równe 1 w kodzie $K(a_s)$.

Tablica układu konwersji kodu dla naszego przykładu przedstawiona jest w tab. 4.

Tab. 4. Tablica układu konwersji kodu automatu-PAY

a_m	a_s	$K_m(a_s)$	$K(Y_t)$	$K(a_s)$	Φ_s	h
a_1	a_2	0	001	001	D_3	1
	a_2	0	011	001	D_3	2
	a_3	1	010	010	D_2	3
a_2	a_3	0	100	010	D_2	4
	a_4	1	101	011	$D_2\, D_3$	5
a_3	a_4	0	110	011	$D_2\, D_3$	6
	a_5	1	001	100	D_1	7
a_4	a_1	0	010	000	-	8
	a_5	1	001	100	D_1	9
a_5	a_4	0	100	011	$D_2\, D_3$	10

6. Zaprojektowanie układu logicznego realizującego automat-PAY. Układ **P** realizowany jest za pomocą systemu (5). Dla naszego przykładu z tab. 3. można wyprowadzić następujące funkcje:

$$z_1 = \overline{Q_1}\,\overline{Q_2}Q_3 + \overline{Q_1}Q_2\,\overline{Q_3}\,\overline{x_3} + Q_1\,\overline{Q_2}\,\overline{Q_3}\,;$$

$$z_2 = \overline{Q_1}\,\overline{Q_2}\,\overline{Q_3}\,\overline{x_1} + Q_1\,\overline{Q_2}Q_3\,\overline{x_3} + \overline{Q_1}Q_2Q_3x_1\,;$$

$$z_3 = \overline{Q_1}\,\overline{Q_2}\,\overline{Q_3}x_1 + \overline{Q_1}\,\overline{Q_2}\,\overline{Q_3}\,\overline{x_2} + \overline{Q_1}\,\overline{Q_2}Q_3\,\overline{x_2} + \overline{Q_1}Q_2\,\overline{Q_3}x_3 + \overline{Q_1}Q_2Q_3\,\overline{x_1}\,;$$

$$\tau_1 = \overline{Q_1}\,\overline{Q_2}\,\overline{Q_3}\,\overline{x_1}x_2 + \overline{Q_1}\,\overline{Q_2}Q_3\,\overline{x_2} + \overline{Q_1}\,\overline{Q_2}\,\overline{Q_3}x_3 + \overline{Q_1}Q_2Q_3\,\overline{x_1}\,.$$

Układ konwersji kodu **CC** implementowany jest za pomocą funkcji (6). Dla naszego przykładu można je wyprowadzić z tab. 4.:

$$D_1 = \tau_1\,\overline{z_1}\,\overline{z_2}z_3\,;$$

$$D_2 = \tau_1 \overline{z_1} z_2 \overline{z_3} + \overline{\tau_1} z_1 \overline{z_2 z_3} + \tau_1 \overline{z_1} z_2 z_3 + \overline{\tau_1} z_1 z_2 \overline{z_3} + \overline{\tau_1 z_1 z_2 z_3};$$

$$D_3 = \overline{\tau_1 z_1 z_2 z_3} + \overline{\tau_1} z_1 z_2 z_3 + \tau_1 z_1 \overline{z_2} z_3 + \overline{\tau_1} z_1 z_2 \overline{z_3} + \overline{\tau_1} z_1 \overline{z_2 z_3}.$$

4. ZAKOŃCZENIE

Metoda zaprezentowana w referacie pozwala na zmniejszenie zasobów sprzętowych wymaganych do realizacji automatu skończonego z wyjściami typu Mealy'ego. Metoda z wielokrotnym kodowaniem stanów wewnętrznych, w pewnych warunkach, zapewnia zmniejszenie liczby wyjść układu **P** co prowadzi do zmniejszenia kosztów realizowanego układu.

Przeprowadzone badania, w oparciu o wyniki pracy [3], ukazały, że proponowana metoda zawsze jest lepsza od standardowej. Zysk osiąga 8%-10% i występuje, gdy sieć działań projektowanego automatu posiada wysoką liczbę bloków wykonawczych.

LITERATURA

[1] S. Baranov: *Logic Synthesis for Control Automata*, Kluwer Academic Publishers, 1994

[2] A. Barkalov: *Structures of the multilevel circuits of microprogram automata on PLA*, Cybernetics and system analysis, No. 4, 1994, pp. 22-29

[3] A. Barkalov: *Development of formal methods of structural synthesis of compositional automata*, Donetsk: DonTSY, 1994

[4] T. Łuba, et. al.: *Synteza układów cyfrowych*, Wydawnictwa Komunikacji i Łączności, 2003