

Synteza mikroprogramowanych układów sterujących z dekoderni funkcji

mgr inż. Remigiusz Wiśniewski

Mgr inż. Remigiusz Wiśniewski jest absolwentem Uniwersytetu Zielonogórskiego (2003). Ukończył studia o specjalności Inżynieria Komputerowa. W roku 2000 odbył przemysłową praktykę studencką w firmie Aldec Inc. w Stanach Zjednoczonych. Od roku 2003 pracuje jako asystent na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: r.wisniewski@iie.uz.zgora.pl

**prof. 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.barkalov@iie.uz.zgora.pl



Streszczenie

W referacie zaprezentowano metodę umożliwiającą zmniejszenie liczby wyjść oraz funkcji logicznych modułów wewnętrznych układu sterującego poprzez zastosowanie dekodera funkcji. Dodatkowy układ może zostać zrealizowany z wykorzystaniem dedykowanych bloków pamięci, co znacznie pozwala zmniejszyć liczbę elementów logicznych matrycy FPGA. Idea metody została zilustrowana przykładem. Pokazano wszystkie kroki, które są niezbędne do realizacji układu z wykorzystaniem proponowanej metody. Szczegółowe badania wykazały skuteczność metody (zmniejszenie liczby wykorzystanych bloków logicznych docelowego układu FPGA o około 36%).

Abstract

In the article a new method of synthesis of compositional microprogram control unit (CMCU) is presented. The method is based on the modification in the traditional solutions. Application of an additional block - function decoder - permits to reduce the number of logic blocks used for implementation of the CMCU on FPGA. All steps required in order to synthesize CMCU with function decoder are shown. Investigations conducted by authors have shown that the proposed method permits to decrease the FPGA area used for implementation of the control unit up to 36% compared with traditional methods.

Słowa kluczowe: mikroprogramowane układy sterujące, dekoderni funkcji, programowalne matryce FPGA.

Keywords: Compositional Microprogram Control Unit, Function Decoder, Field-Programmable Gate Arrays (FPGAs).

1. Wstępn

Jednostka sterująca jest ważną częścią projektowanego systemu cyfrowego [1,2,4,5]. Standardowa metoda implementacji jednostki sterującej w postaci skończonego automatu stanów często pochłania zasoby układów programowalnych, które mogą być w efektywniejszy sposób wykorzystane przez inne bloki projektowanego systemu [4]. Coraz częściej spotykanym rozwiązaniem jest układ mikroprogramowany, w którym zastosowano dekompozycję jednostki sterującej na część zarządzającą (adresującą) oraz pamięć, w której przechowywane są mikroinstrukcje kontrolera [2,3].

Jednym ze sposobów projektowania mikroprogramowanych układów sterujących jest wykorzystanie adresu mikroinstrukcji jako stanów wewnętrznych mikrokontrolera. Takie podejście często pozwala znacznie zmniejszyć liczbę przerzutników niezbędnych do realizacji układu, jednakże problemem cały czas pozostaje, liczba funkcji logicznych realizowanych przez moduły wewnętrzne sterownika.

W artykule przedstawiono metodę umożliwiającą zmniejszenie liczby wyjść oraz połączeń wewnętrznych pomiędzy poszczególnymi blokami wewnętrznymi sterownika. W tym celu zastosowano dodatkowy układ dekodera funkcji, który może zostać zrealizowany z wykorzystaniem dedykowanych bloków pamięci matrycy FPGA, co pozwala zaoszczędzić zasoby elementów logicznych układów programowalnych.

W referacie przedstawiono tylko najważniejsze definicje dotyczące mikroprogramowanych układów sterujących. Szczegóły dotyczące projektowania jednostek mikroprogramowanych można znaleźć w literaturze [1,5] oraz licznych artykułach [3,6,7].

2. Podstawowe definicje

Niech dany będzie algorytm sterowania, przedstawiony za pomocą sieci działań Γ [2], składającej się ze zbioru bloków operacyjnych $B = \{b_1, \dots, b_K\}$, zbioru bloków decyzyjnych (warunkowych) $X = \{x_1, \dots, x_L\}$ oraz zbioru połączeń E . Każdy blok operacyjny $b_k \in B$ zawiera mikroinstrukcję, będącą zbiorem mikrooperacji $Y(b_k) \subseteq Y$, gdzie $Y = \{y_1, \dots, y_N\}$ jest zbiorem niepustych mikrooperacji. Sieć działań zawiera węzeł początkowy b_0 oraz węzeł końcowy b_E .

Definicja 1. Łańcuch bloków operacyjnych w sieci działań Γ to skończona sekwencja węzłów operacyjnych $\alpha_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$, takich, że dla każdej pary sąsiednich bloków wektora α_g istnieje połączenie $\langle b_{gi}, b_{gi+1} \rangle \in E$, gdzie i jest numerem bloku wchodzącego w skład wektora $\alpha_g (i=1, \dots, F_g-1)$.

Definicja 2. Blok $b_q \in B$ jest wejściem łańcucha α_g , jeśli istnieje połączenie $\langle b_p, b_q \rangle \in E$, że b_p jest blokiem początkowym lub blokiem decyzyjnym sieci działań Γ , który nie należy do łańcucha α_g .

Definicja 3. Blok $b_q \in B$ jest wyjściem łańcucha α_g , jeśli istnieje krawędź $\langle b_q, b_r \rangle \in E$, taka, że b_r jest blokiem nienależącym do α_g .

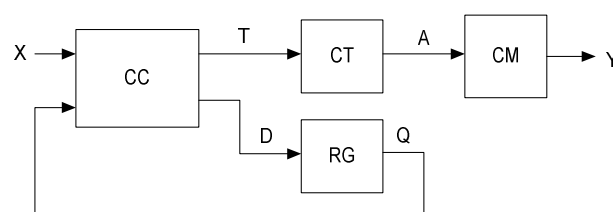
Niech D^g będzie zbiorem bloków operacyjnych wchodzących w skład łańcucha α_g . Wówczas $C = \{\alpha_1, \dots, \alpha_G\}$ jest zbiorem wszystkich łańcuchów wchodzących w skład sieci działań Γ , spełniających warunek:

$$\begin{aligned} D^g \cap D^q &= \emptyset (g \neq q; g, q \in \{1, \dots, G\}); \\ B &= D^1 \cup D^2 \cup \dots \cup D^G; \\ D^g &\neq \emptyset (g = 1, \dots, G). \end{aligned} \quad (1)$$

Jeśli adres mikroinstrukcji zostanie zakodowany z wykorzystaniem naturalnego kodu binarnego dla każdego łańcucha $\alpha_g \in C$:

$$A(b_{gi+1}) = A(b_{gi}) + 1 (i = 1, \dots, F_{g-1}), \quad (2)$$

gdzie $A(b_g)$ jest adresem mikroinstrukcji generowanej w bloku operacyjnym $b_g \in B$. W tym przypadku sieć działań Γ może zostać zrealizowana za pomocą mikroprogramowanego układu sterującego o strukturze podstawowej (rys. 1).



Rys. 1. Mikroprogramowany układ sterujący o strukturze podstawowej
Fig. 1. Compositional microprogram control unit with base structure

W układzie zilustrowanym poprzez rys. 1. blok kombinacyjny CC oraz rejestr RG tworzą uproszczony skończony automat stanów. Automat ten jest odpowiedzialny za wyznaczanie funkcji wzbudzeń dla licznika CT. Licznik oraz pamięć układu CM stanowią blok przechowujący oraz generujący mikroinstrukcje. Struktura oraz sposoby projektowania mikroprogramowanego układu sterującego są szczegółowo opisane w literaturze [2,3].

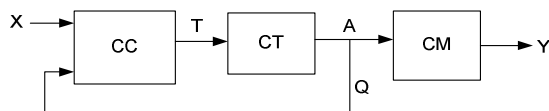
3. Mikroprogramowany układ sterujący z identyfikacją wyjść

W celu zredukowania liczby bloków wewnętrznych kontrolera, adres mikroinstrukcji (wyjście licznika) może być wykorzystane do kodowania wewnętrznych stanów sterownika. W układzie przedstawionym na rys. 2. dodatkowo zastosowano kodowanie mikroinstrukcji, dzięki czemu kod stanu kontrolera stanowi część adresu mikroinstrukcji. Dzięki temu układ kombinacyjny generuje jedynie funkcje wzbudzeń dla licznika:

$$T = f(X, Q), \quad (3)$$

gdzie X oznacza zbiór zmiennych warunkowych, Q – zbiór zmiennych określających stan mikrokontrolera ($Q \subseteq A$).

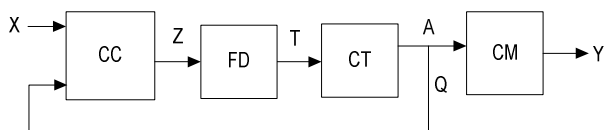
Szczegółowy sposób kodowania mikroinstrukcji oraz funkcjonalność mikroprogramowanego układu sterującego o adresowaniu wspólnym opisany został w [8].



Rys. 2. Mikroprogramowany układ sterujący z identyfikacją wyjść
Fig. 2. Compositional microprogram control unit with outputs identification

4. Idea proponowanej metody

Układ przedstawiony na rys. 2. pozwala zmniejszyć liczbę wykorzystanych bloków logicznych poprzez wykorzystanie części adresu mikroinstrukcji. W niniejszym artykule proponujemy zastosowanie dekodera funkcji. Funkcje generowane przez układ kombinacyjny zostają zakodowane, które następnie są dekodowane przez dodatkowy blok (rys. 3.).



Rys. 3. Mikroprogramowany układ sterujący z dekoderni funkcji
Fig. 3. Compositional microprogram control unit with function decoder

Układ przedstawiony na rys. 3. działa w następujący sposób: układ kombinacyjny, dekoderni funkcji oraz licznik tworzą uproszczony skończony automat stanów, przy czym stany wewnętrzne mikrokontrolera są kodowane z wykorzystaniem części adresu mikroinstrukcji $Q \subseteq A$. Funkcje generowane przez blok kombinacyjny zostają zakodowane (zmienna Z). Są one przetwarzane przez dekoderni funkcji, a następnie podawane na wejście licznika (zmienna T). Licznik wyznacza adres A odpowiedniej mikroinstrukcji, która jest przechowywana w pamięci sterownika. Taki sposób realizacji mikroprogramowanego układu sterującego umożliwia znacznie zmniejszyć liczbę elementów logicznych matrycy FPGA, jakie są niezbędne do implementacji, w porównaniu do rozwiązania przedstawionego na rysunku 2. Zarówno pamięć, jak i dekoderni funkcji mogą zostać zrealizowane z wykorzystaniem dedykowanych bloków pamięci układu FPGA.

5. Synteza mikroprogramowanego układu sterującego z dekoderni funkcji

Realizacja mikroprogramowanego układu sterującego z identyfikacją wyjść oraz dekoderni funkcji przebiega w następujących etapach:

1. **Utworzenie sieci łańcuchów bloków operacyjnych.** Na podstawie początkowej sieci działań wyznaczony zostaje zbiór łańcuchów bloków operacyjnych. Następnie określone zostają wszystkie wejścia oraz wyjścia dla poszczególnych łańcuchów. Na tej podstawie utworzona zostaje sieć łańcuchów bloków operacyjnych.
2. **Zakodowanie mikroinstrukcji oraz określenie zawartości pamięci mikroprogramowanego układu sterującego.** Adresy mikroinstrukcji zostają określone w taki sposób, aby wyjścia poszczególnych łańcuchów bloków operacyjnych były możliwe do jednoznacznego zidentyfikowania z wykorzystaniem jak najmniejszej liczby bitów. Szczegółowy opis kodowania mikroinstrukcji przedstawiony został w [8].
3. **Utworzenie tabeli przejść mikroprogramowanego układu sterującego, kodowanie stanów oraz wyznaczenie funkcji wzbudzeń dekodera.** W tym kroku utworzona zostaje tabela przejść pomiędzy poszczególnymi łańcuchami bloków operacyjnych. Pod uwagę brane są tylko te łańcuchy, które nie są połączone z końcowym blokiem w sieci działań. Wyjścia poszczególnych łańcuchów identyfikowane są na podstawie minimalnej liczby bitów, wchodzących w skład funkcji Q . Na podstawie tabeli przejść wyznaczona zostaje funkcja wzbudzeń Z dla układu dekodera funkcji.
4. **Utworzenie tabeli prawdy dla dekodera funkcji.** Na tym etapie należy utworzyć tabelę prawdy dla dekodera funkcji, a następnie na jej podstawie wyznaczyć funkcję wzbudzeń dla licznika. Wejściem układu dekodera jest funkcja Z , natomiast wyjściem - funkcja T .
5. **Implementacja układu z wykorzystaniem matrycy programowalnych FPGA.** W przypadku struktur FPGA zarówno pamięć sterownika jak i blok dekodera funkcji mogą zostać zaimplementowane na dwa sposoby. Pierwsza możliwość to realizacja z wykorzystaniem dedykowanych bloków pamięci matrycy FPGA. Takie rozwiązanie pozwala znacznie zmniejszyć liczbę wykorzystanych bloków logicznych. Drugi sposób to implementacja z wykorzystaniem bloków logicznych FPGA. Opcja ta jest stosowana zazwyczaj tylko wtedy, gdy rozmiar pamięci lub dekodera funkcji przekracza rozmiar dostępnego miejsca w pamięciach dedykowanych.

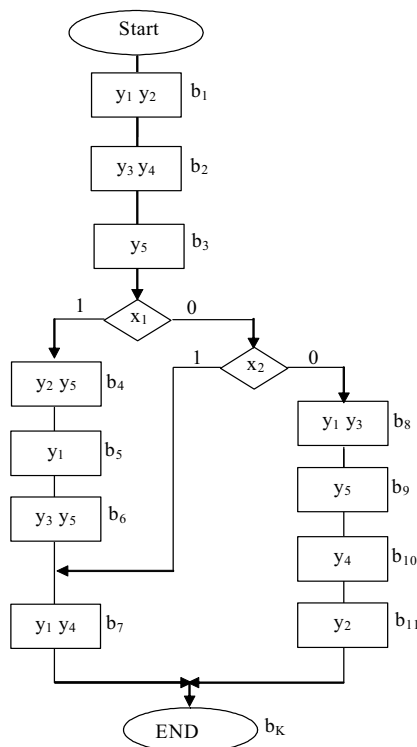
6. Przykład syntezy sterownika z wykorzystaniem proponowanej metody

Sposób projektowania mikroprogramowanego układu sterującego z dekoderni funkcji zostanie zilustrowany przykładem. W tym celu zaproponowana zostanie hipotetyczny algorytm sterowania Γ_1 , przedstawiony na rys. 4. Na tej podstawie przygotowany zostanie mikroprogramowany układ sterujący U_1 .

W przedstawionej sieci działań występują trzy łańcuchy bloków operacyjnych: $C = \{\alpha_1, \alpha_2, \alpha_3\}$. Łańcuch α_1 zawiera trzy bloki: $\alpha_1 = \langle b_1, b_2, b_3 \rangle$, łańcuch α_2 cztery bloki operacyjne $\alpha_2 = \langle b_4, b_5, b_6, b_7 \rangle$, podobnie jak ostatni łańcuch $\alpha_3 = \langle b_8, b_9, b_{10}, b_{11} \rangle$. W prezentowanym przykładzie istnieją w sumie cztery wejścia łańcuchów operacyjnych, przy czym:

- łańcuch α_1 posiada jedno wejście: I_1^1 którym jest blok b_1 ,
- łańcuch α_2 posiada dwa wejścia: $I_2^1 = b_4$, oraz $I_2^2 = b_7$,
- łańcuch α_3 posiada jedno wejście: $I_3^1 = b_8$.

Kolejny krok to kodowanie adresu mikroinstrukcji oraz określenie zawartości pamięci układu sterującego. W prezentowanym przykładzie adres mikroinstrukcji zostanie zakodowany zgodnie z algorytmem przedstawionym w [8]. Wynika stąd, że dwa bity są potrzebne do rozpoznania każdego łańcucha: $K(\alpha_1)=00$, $K(\alpha_2)=01$, $K(\alpha_3)=10$.



Rys. 3. Algorytm sterowania Γ_1
Fig. 3. Flow-chart Γ_1

Kodowanie bloków wchodzących w skład łańcuchów nie jest istotne, dlatego też w tym przypadku zostanie wykorzystany naturalny kod binarny. Ostatecznie adres mikroinstrukcji jest wyznaczany na podstawie konkatenacji:

$$A(b_i) = K(\alpha_g) * K(b_i), \quad (4)$$

gdzie $A(b_i)$ oznacza adres mikroinstrukcji, $K(\alpha_g)$ – kod łańcucha, $K(b_i)$ – kod bloku operacyjnego wchodzącego w skład łańcucha α_g . Przykładowo:

$$\begin{aligned} A(b_1) &= K(\alpha_1) * K(b_1) = 0000 \text{ (blok początkowy sieci działań),} \\ &\dots \\ A(b_3) &= K(\alpha_1) * K(b_3) = 0010 \text{ (wyjście łańcucha } \alpha_1), \\ &\dots \\ A(b_{11}) &= K(\alpha_3) * K(b_{11}) = 1011. \end{aligned}$$

Kolejny etap to określenie zawartości tabeli przejść mikroprogramowanego układu sterującego. W prezentowanym przykładzie jedynie wyjście łańcucha α_1 nie jest połączone z końcowym blokiem sieci działań, wobec czego wystarczy jeden bit do rozpoznania wyjścia (w tym przypadku wykorzystany zostanie najstarszy bit kodu) $SA(O_1) = 0010$ (tab. 1).

Tab. 1. Tabela przejść mikroprogramowanego układu sterującego U_1
Tab. 1. Table of transitions of the CMCU U_1

O_g	$SA(O_g)$	X_h	I_q^j	$K(I_q^j)$	Z	h
O_1	0	x_1	I_2^1	01	z_1	1
		$\overline{x_1}x_2$	I_2^2	10	z_2	2
		$\overline{x_1}\overline{x_2}$	I_3^1	11	$z_1 z_2$	3

Na podstawie tabeli przejść określone mogą zostać równania wzbudzeń dla dekodera funkcji, przykładowo:

$$z_1 = \overline{Q} * (x_1 + \overline{x_1 x_2}). \quad (5)$$

W następnym kroku określona zostaje tabela prawdy dla dekodera funkcji. Na podstawie kodu wejścia danego łańcucha Z dekodowana jest wartość dla licznika T (tab. 2.).

Tab. 2. Zawartość dekodera funkcji dla układu U_1
Tab. 2. Content of the function decoder

I_q^j	$K(I_q^j)$	$A(I_q^j)$	T	h
I_1^1	00	0000	--	1
I_2^1	01	0100	t_2	2
I_2^2	10	0111	$t_2 t_3 t_4$	3
I_3^1	11	1100	$t_1 t_2$	4

Dekoder funkcji może zostać zaimplementowany na dwa sposoby: z wykorzystaniem bloków logicznych lub dedykowanych bloków pamięci. W pierwszym przypadku wyznaczone zostają dodatkowo równania wzbudzeń dla licznika, przykładowo:

$$t_1 = z_1 * z_2. \quad (6)$$

Ostatni etap to logiczna synteza oraz implementacja układu w matrycach programowalnych. Należy tu zwrócić uwagę, że zarówno blok dekodera funkcji, jak i pamięć sterownika mogą zostać zrealizowane z wykorzystaniem dedykowanych bloków pamięci układów FPGA, co znacznie pozwala zmniejszyć liczbę wykorzystanych bloków logicznych w porównaniu do rozwiązań tradycyjnych.

7. Podsumowanie

W artykule przedstawiona została nowa metoda projektowania mikroprogramowanych układów sterujących z wykorzystaniem dekodera funkcji. Zastosowanie dodatkowego bloku umożliwia zmniejszenie połączeń wewnętrznych sterownika, co w efekcie pozwala zredukować liczbę elementów logicznych, jakie są niezbędne do implementacji układu.

Szczegółowe badania przeprowadzone przez autorów potwierdzają skuteczność proponowanego rozwiązania. W porównaniu do tradycyjnego mikroprogramowanego układu sterującego proponowana metoda umożliwia zmniejszenie wykorzystanych zasobów logicznych układu FPGA średnio o 36%.

8. Literatura

- [1] Baranov S.: Logic Synthesis for Control Automata, Kluwer Academic Publishers, 1994.
- [2] Barkalov A.A., Palagin A.V.: Synthesis of Microprogram Control Units, IC NAS of Ukraine, Kiev, Ukraine, 1997.
- [3] Barkalov A.A.: Synteza jednostek sterujących w strukturach programowalnych, KNWS'05, Oficyna Wydawnicza UZ, Zielona Góra, 2005.
- [4] DeMicheli G.: Synthesis and Optimization of Digital Circuits, McGraw Hill, New York, 1994.
- [5] Łuba T (Praca zbiorowa pod redakcją prof. Tadeusza Łuby): Synteza układów cyfrowych, WKŁ, Warszawa, 2003.
- [6] Wisniewski R.: Częściowa rekonfiguracja mikroprogramowanych układów sterujących implementowanych z wykorzystaniem struktur FPGA, OWD 2005, Archiwum Konferencji PTETiS, Wisła, 2005.
- [7] Wisniewski R.: Projektowanie układów mikroprogramowanych z wykorzystaniem wbudowanych bloków pamięci w matrycach programowalnych, KNWS'05, Oficyna Wydawnicza UZ, Zielona Góra, 2005.
- [8] Wiśniewski R., Barkalov A., Titarenko L., Optimization of address circuit of CMCU, EWDTW '06. Kharkov, 2006, pp. 167—170

Title: Synthesis of CMCU with function decoder.

Artykuł recenzowany