

Remigiusz WIŚNIEWSKI, Alexander BARKALOV
 UNIwersytet Zielonogórski, INSTYTUT Informatyki i Elektrotechniki

Synteza mikroprogramowanych układów sterujących z identyfikacją wyjść oraz dekodery 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

Streszczenie

W artykule zaprezentowana zostanie metoda umożliwiająca zmniejszenie liczby wyjść oraz funkcji logicznych modułów wewnętrznych układu sterującego poprzez zastosowanie identyfikacji wyjść łańcuchów bloków operacyjnych. Ponadto wprowadzony zostanie dodatkowy blok dekodera funkcji, który może zostać zrealizowany z wykorzystaniem dedykowanych bloków pamięci, co znacznie pozwala zmniejszyć liczbę elementów logicznych matrycy FPGA. Szczegółowe badania przeprowadzone przez autorów potwierdzają skuteczność proponowanej metody. W porównaniu ze standardowym sposobem projektowania układów mikroprogramowanych, rozwiązanie wykorzystujące dekodery funkcji pozwala zmniejszyć liczbę wykorzystanych bloków logicznych średnio o 36%.

Słowa kluczowe: mikroprogramowany układ sterujący, dekodery funkcji, programowalny układ FPGA.

Synthesis of compositional microprogram control units with outputs identification and function decoder

Abstract

A new synthesis method of compositional microprogram control unit (CMCU) is presented in the article. 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 will be shown. Detailed 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 solutions.

Keywords: Compositional Microprogram Control Unit (CMCU), function decoder, Field-Programmable Gate Arrays (FPGAs).

1. Wstęp

Jednostka sterująca jest ważną częścią 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].

W artykule przedstawiono metodę umożliwiającą zmniejszenie liczby wyjść oraz połączeń wewnętrznych pomiędzy poszczególnymi blokami wewnętrznymi mikroprogramowanego układu sterującego. W tym celu zastosowano dodatkowy układ dekodera funkcji, który może zostać zrealizowany z wykorzystaniem dedykowanych bloków pamięci matrycy FPGA,

Prof. dr hab. inż. Alexander A. BARKALOV

W latach 1976-1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki Donieckiej. Współpracował aktywno z Instytutem 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.



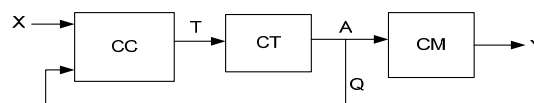
e-mail: a.barkalov@iie.uz.zgora.pl

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. Mikroprogramowany układ sterujący

Jednostka sterująca może zostać zrealizowana jako mikroprogramowany układ sterujący [2, 8]. Podstawową cechą takiego systemu jest podział sterownika na część zarządzającą oraz część przechowującą i generującą mikrooperacje. Przykładowy mikroprogramowany układ sterujący z identyfikacją wyjść przedstawiony został na rys. 1.



Rys. 1. Mikroprogramowany układ sterujący z identyfikacją wyjść
 Fig. 1. Compositional microprogram control unit with outputs identification

W układzie zilustrowanym przez rys. 1 blok kombinacyjny CC odpowiedzialny jest za wyznaczenie funkcji wzbudzeń T dla licznika CT. Licznik oraz pamięć układu CM stanowią blok przechowujący oraz generujący mikroinstrukcje. W układzie dodatkowo zastosowano kodowanie mikroinstrukcji, dzięki czemu kod stanu Q kontrolera stanowi część adresu A mikroinstrukcji:

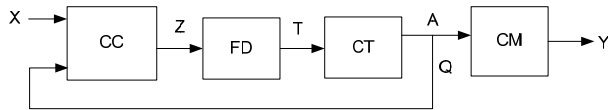
$$T = f(X, Q), \quad (1)$$

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

Podstawową zaletą układu zrealizowanego jako mikroprogramowany układ sterujący jest elastyczność w wyborze sposobu implementacji modułu pamięci sterownika. Blok CM może zostać zrealizowany zarówno w dedykowanych blokach pamięci, jak i z wykorzystaniem elementów logicznych matrycy FPGA. Ponadto implementacja w dedykowanych blokach pamięci wiąże się z możliwością wykorzystania częściowej rekonfiguracji sterownika, co znacznie skraca czas projektowania kolejnej wersji układu sterującego. Wymieniana jest wówczas jedynie zawartość dedykowanych bloków pamięci matrycy FPGA, pozostałe elementy pozostają bez zmian. Idea częściowej rekonfiguracji została bardzo szczegółowo przedstawiona w [6].

3. Idea proponowanej metody

Układ przedstawiony na rys. 2 pozwala zmniejszyć liczbę wykorzystanych bloków logicznych poprzez wykorzystanie części adresu mikroinstrukcji. W niniejszym artykule zaproponowano zastosowanie dekodera funkcji. Funkcja wzbudzeń dla licznika zostaje zakodowana z wykorzystaniem minimalnej, niezbędnej liczby bitów (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 $Q \subseteq A$. Funkcja wzbudzeń generowana przez blok kombinacyjny zostaje zakodowana (zmienna Z). Jest ona przetwarzana przez dekoderni funkcji, a następnie podawana 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 matryc FPGA, które 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.

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

1. Tworzenie sieci łańcuchów bloków operacyjnych.

Na podstawie początkowej sieci działań wyznaczony zostaje zbiór łańcuchów bloków operacyjnych. Proces ten polega na zastąpieniu ciągu bloków operacyjnych jednym blokiem, zwanym *łańcuchem bloków operacyjnych* [2, 3, 6, 7, 8]. 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 referacie [8].

3. Tworzenie 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 zidentyfikowane 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. Tworzenie tabeli prawdy dla dekodera funkcji.

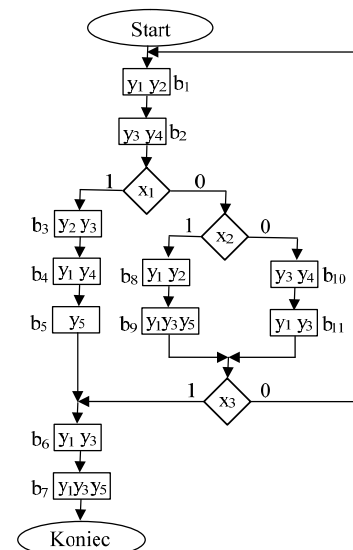
Na tym etapie należy utworzyć tabelę prawdy dla dekodera funkcji, a następnie na jej podstawie wyznaczyć funkcję wzbudzeń dla licznika. Wejściem układu dekodera jest funkcja Z , natomiast wyjściem - funkcja T .

5. Implementacja układu z wykorzystaniem matryc 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 matryc 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.

4. Przykład syntezy mikroprogramowanego układu sterującego z identyfikacją wyjść oraz dekoderni funkcji

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 .



Rys. 4. Algorytm sterowania Γ_1
Fig. 4. Flow-chart Γ_1

W przedstawionej sieci działań występują cztery łańcuchy bloków operacyjnych: $C = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. Łańcuch α_1 zawiera dwa bloki operacyjne $\alpha_1 = \langle b_1, b_2 \rangle$, łańcuch α_2 pięć bloków $\alpha_2 = \langle b_3, b_4, b_5, b_6, b_7 \rangle$, natomiast łańcuchy α_3 i α_4 po dwa bloki: $\alpha_3 = \langle b_8, b_9 \rangle$, $\alpha_4 = \langle b_{10}, b_{11} \rangle$. W prezentowanym przykładzie istnieje w sumie pięć wejść łańcuchów operacyjnych, przy czym łańcuch α_1 posiada jedno wejście: I_1^1 , którym jest blok b_1 , łańcuch α_2 ma dwa wejścia: $I_2^1 = b_3$ oraz $I_2^2 = b_6$, łańcuch α_3 posiada jedno wejście

$I_3^1 = b_8$, natomiast łańcuch α_4 ma jedno wejście: $I_4^1 = b_{10}$. Każdy łańcuch może posiadać tylko jedno wyjście. Wyjściem O_1 pierwszego łańcucha jest blok b_2 . Analogicznie wyjściem O_2 drugiego łańcucha jest blok b_7 , trzeciego $O_3 = b_9$, zaś czwartego $O_4 = b_{11}$.

Kolejny krok to kodowanie adresu mikroinstrukcji oraz określenie zawartości pamięci układu sterującego. Adres mikroinstrukcji zostanie zakodowany zgodnie z algorytmem przedstawionym w [8]. Wynika stąd, że potrzebne są dwa bity do rozpoznania wyjścia każdego łańcucha. Dla przedstawionego przykładu będą to najstarsze bity adresu poszczególnych wyjść:

$$\begin{aligned} K(O_1) &= \mathbf{0001}, \\ K(O_2) &= \mathbf{0110}, \\ K(\alpha_3) &= \mathbf{1000}, \\ K(\alpha_4) &= \mathbf{1100}. \end{aligned}$$

W następnym etapie określona zostaje zawartość tabeli przejść mikroprogramowanego układu sterującego. W prezentowanym przykładzie jedynie wyjście O_2 łańcucha α_2 nie jest połączone z końcowym blokiem sieci działań, wobec czego pod uwagę brane są wyjścia pozostałych trzech łańcuchów (tab. 1).

Tab. 1. Tabela przejść mikroprogramowanego układu sterującego U_1
Tab. 1. Transitions table of the CMCU U_1

O_g	$SA(O_g)$	X_h	I_q^j	$K(I_q^j)$	Z	h
O_1	00	x_1	I_2^1	001	z_1	1
		$\overline{x_1 x_2}$	I_3^1	011	$z_1 z_2$	2
		$\overline{x_1 x_2}$	I_4^1	100	z_3	3
O_2	01	--	--	--	--	4
O_3	10	x_3	I_2^2	010	z_2	5
		$\overline{x_3}$	I_1^1	000	--	6
O_4	11	x_3	I_2^2	010	z_2	7
		$\overline{x_3}$	I_1^1	000	--	8

W prezentowanej tabeli pole $SA(O_g)$ określa minimalną liczbę bitów, niezbędnych do jednoznacznej identyfikacji wyjścia O_g . Na podstawie tabeli przejść określone zostają równania wzbudzeń dla dekodera funkcji, przykładowo:

$$z_1 = \overline{Q_1} \overline{Q_2} * (\overline{x_1} + \overline{x_1 x_2}). \quad (2)$$

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). Najczęściej dekoderek funkcji jest realizowany z wykorzystaniem dedykowanych bloków pamięci, jednakże należy podkreślić fakt możliwości implementacji tego bloku z wykorzystaniem bloków logicznych matryc FPGA.

Ostatni etap to logiczna synteza oraz implementacja układu w matrycach programalnych. 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.

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	000	0000	--	1
I_2^1	001	0010	t_2	2
I_2^2	010	0101	$t_1 t_3$	3
I_3^1	011	0111	$t_1 t_2 t_3$	4
I_4^1	100	1011	$t_1 t_3 t_4$	5

5. 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%.

6. 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 programalnych, 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 programalnych, 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