

Remigiusz WIŚNIEWSKI

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Synteza mikroprogramowanych układów sterujących ze współdzieleniem kodów z wykorzystaniem dekodera adresów

Mgr inż. Remigiusz WIŚNIEWSKI

Mgr inż. Remigiusz Wiśniewski jest absolwentem Uniwersytetu Zielonogórskiego (2004). Ukończył studia o specjalności Inżynieria Komputerowa. W roku 2000 odbył przemysłową praktykę studencką w firmie Aldec Inc. w Stanach Zjednoczonych. Od roku 2004 pracuje jako asystent na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.



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

Streszczenie

W artykule zaprezentowana została metoda umożliwiająca syntezę mikroprogramowanych układów sterujących ze współdzieleniem kodów niezależnie od rozmiaru adresu mikroinstrukcji sterownika. Metoda bazuje na wykorzystaniu dodatkowego dekodera adresów, który na podstawie kodu łańcucha oraz zakodowanego kodu bloku operacyjnego wyznacza odpowiedni adres mikroinstrukcji. Idea metody została zilustrowana przykładem. Pokazano wszystkie kroki, które są niezbędne do realizacji układu z wykorzystaniem proponowanej metody.

Słowa kluczowe: programowane układy sterujące, współdzielenie kodów, dekodery adresów.

Synthesis of Compositional Microprogram Control Units with Sharing Codes and Address Converter**Abstract**

The method of optimization of logic circuit of compositional microprogram control unit with sharing of the codes is proposed. Method is based on the transformation of the pairs (code of operational linear chain, maximal code of input) in the addresses of the input of operational linear chain. The method of design and example of its application is proposed.

Keywords: Compositional Microprogram Control Unit, Sharing Codes, Address Decoder (Address Transformer).

1. Wstęp

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 metody współdzielenia kodów, gdzie adres mikroinstrukcji jest wyznaczany na podstawie kodu łańcucha oraz kodu bloku operacyjnego. Taki sposób realizacji sterownika pozwala zmniejszyć ilość wykorzystanych bloków logicznych docelowego układu. Rozwiązanie to ma jednak poważne ograniczenie: rozmiar adresu mikroinstrukcji musi być równy łącznej liczbie bitów, potrzebnej do zakodowania łańcucha oraz bloku operacyjnego [2].

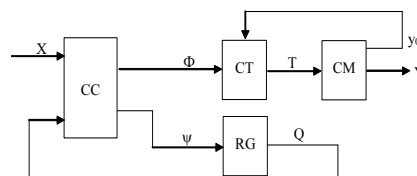
W artykule przedstawiono metodę umożliwiającą wykorzystanie sposobu projektowania układów sterujących ze współdzieleniem kodów niezależnie od rozmiaru adresu mikroinstrukcji. W tym celu zastosowano dodatkowy układ dekodera adresów,

który jest odpowiedzialny za wyznaczenie funkcji wzbudzeń dla licznika na podstawie kodu stanu generowanego przez rejestr RG oraz wartości wyznaczanych przez blok CC.

W referacie pominięto podstawowe definicje związane z mikroprogramowanymi układami sterującymi, gdyż są one szczegółowo opisane w literaturze [1, 5] oraz w licznych artykułach [3, 6, 7].

2. Mikroprogramowany układ sterujący o strukturze podstawowej

Jednostka sterująca może zostać zrealizowana jako mikroprogramowany układ sterujący [2, 8]. Podstawową cechą takiego systemu jest podział mikrokontrolera na część zarządzającą oraz część przechowującą i generującą mikrooperacje. Mikroprogramowany układ sterujący o strukturze podstawowej przedstawiony został na rys. 1.

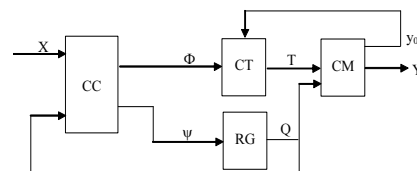


Rys. 1. Mikroprogramowany układ sterujący o strukturze podstawowej
Fig. 1. Compositional Microprogram Control Unit with base structure

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 ze współdzieleniem kodów

Jednym ze sposobów projektowania układów mikroprogramowanych jest wykorzystanie metody współdzielenia kodów. Wówczas adres mikroinstrukcji jest wyznaczany na podstawie kodu łańcucha bloku operacyjnego oraz kodu bloku operacyjnego.



Rys. 2. Mikroprogramowany układ sterujący ze współdzieleniem kodów
Fig. 2. Compositional Microprogram Control Unit with Sharing Codes

Na rys. 2 przedstawiono mikroprogramowany układ sterujący ze współdzieleniem kodów [2]. Sterownik działa w podobny sposób jak w przypadku układu o strukturze podstawowej. Blok CC oraz rejestr RG tworzą uproszczony automat stanów. Jednakże tym razem licznik CT wyznacza jedynie adres mikroinstrukcji dla kolejnych bloków operacyjnych w danym łańcuchu. Kod samego łańcucha wyznaczany jest na podstawie funkcji generowanej przez rejestr.

Taki sposób realizacji sterownika pozwala zmniejszyć ilość wykorzystanych bloków logicznych docelowego układu, w porównaniu do implementacji układu o strukturze podstawowej. Wynika to z faktu, że do wyznaczenia adresu mikroinstrukcji wykorzystany jest także kod wyznaczany przez rejestr RG. Pozwala to znacznie zmniejszyć rozmiar funkcji Z generowanej przez blok CC, a w konsekwencji rozmiar licznika CT.

Metoda współdzielenia kodów ma jednak istotne ograniczenie. Jeśli przez R_1 oznaczony zostanie rozmiar adresu generowanego przez licznik, przez R_2 rozmiar adresu generowanego przez rejestr, natomiast przez R_3 - minimalną liczbę bitów, jaka jest potrzebna na zaadresowanie mikroinstrukcji, to warunek

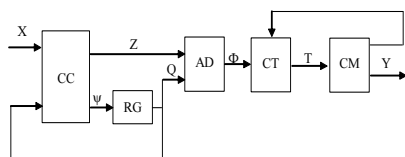
$$R_1 + R_2 = R_3 \quad (1)$$

powinien zostać spełniony. W innym przypadku rozmiar pamięci układu zostanie drastycznie zwiększona. Problem jest o tyle istotny, iż jest to wzrost wykładniczy. Przykładowo, gdy wartość $R_1 + R_2$ jest większa od R_3 o 1, wówczas rozmiar pamięci zwiększy się dwukrotnie, (gdy o 2 - aż czterokrotnie, itd.).

W kolejnym rozdziale zaprezentowana zostanie metoda umożliwiająca wykorzystanie współdzielenia kodów, gdy warunek (1) nie jest spełniony.

4. Idea proponowanej metody

Na rys. 3 przedstawiony został mikroprogramowany układ sterujący z dekoderni adresów (AD). Blok ten ma za zadanie wyznaczenie odpowiednich wartości dla licznika na podstawie stanu, w jakim znajduje się układ (wyjście bloku RG) oraz dodatkowej informacji, generowanej przez układ kombinacyjny CC.



Rys. 3. Mikroprogramowany układ sterujący z dekoderni adresów
Fig. 3. Compositional Microprogram Control Unit with Address Decoder

Układ przedstawiony na rys. 3 funkcjonuje w następujący sposób: blok CC oraz rejestr RG tworzą uproszczony automat stanów, umożliwiający przejścia sterownika pomiędzy poszczególnymi łańcuchami bloków operacyjnych [3]. Dekoder AD wyznacza adresy wejść poszczególnych łańcuchów. Informacja ta jest z kolei pobierana przez licznik CT, który generuje adresy mikroinstrukcji kolejnych bloków łańcucha. Mikroinstrukcje te są przechowywane w pamięci CM mikroprogramowanego układu sterującego.

Dekoder AD ma za zadanie wyznaczyć adresy wejść poszczególnych łańcuchów bloków operacyjnych. Dlatego też oczywiste jest, że funkcja Z, będzie generowana przez blok CC tylko wtedy, gdy dany łańcuch ma więcej niż jedno wejście. W przeciwnym przypadku do przekazania wszystkich niezbędnych informacji (czyli adresu pierwszego wejścia odpowiedniego łańcucha) wystarczy wartość generowana przez rejestr RG.

Podstawową zaletą tak zrealizowanego układu sterującego jest możliwość wykorzystania metody współdzielenia kodów nawet w przypadku, gdy warunek (1) nie jest spełniony. Ponadto należy zwrócić uwagę na fakt, że w porównaniu do rozwiązań tradycyjnych, proponowana metoda znacznie zmniejsza ilość wyjść generowanych przez blok CC, ponieważ funkcja Z przechowuje tylko wartości wejść danego łańcucha.

Proponowane rozwiązanie posiada także istotną wadę. Jest nią konieczność implementacji dekodera adresów. Dlatego też metoda ma sens tylko w przypadku, gdy ogólna liczba bloków logicznych wykorzystanych do implementacji sterownika jest mniejsza niż w przypadku struktur tradycyjnych.

5. Synteza mikroprogramowanego układu sterującego z dekoderni adresów

Synteza mikroprogramowanego układu sterującego z dekoderni adresów może zostać podzielona na następujące etapy:

1. Utworzenie łańcuchów bloków operacyjnych oraz utworzenie pamięci mikroprogramowanego układu sterującego. Ten krok został szczegółowo opisany w [Barkalov, 2002]. Poszczególne mikroinstrukcje kodowane są z wykorzystaniem naturalnego kodu binarnego (NKB).

2. Kodowanie wejść łańcuchów bloków operacyjnych. W tym celu należy znaleźć łańcuch posiadający najwięcej wejść. Niech parametr M_4 oznacza ilość wejść w tym łańcuchu. Wówczas ilość bitów potrzebnych do zakodowania zmiennej Z będzie równa:

$$R_4 = \lceil \log_2 M_4 \rceil \quad (2)$$

Następnie należy zakodować wejścia poszczególnych łańcuchów. Jeśli dany łańcuch ma mniej wejść niż M_4 , wówczas starsze bity kodu powinny mieć wartość 0.

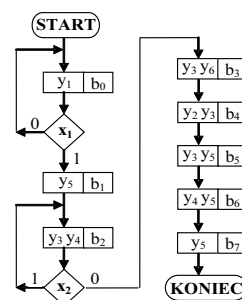
3. Utworzenie tabeli przejść mikroprogramowanego układu sterującego. Jest ona niezbędna do określenia funkcji wzbudzeń dla rejestru (Ψ) oraz funkcji przechowującej kody wejść poszczególnych łańcuchów (Z).

4. Utworzenie tabeli dla układu dekodera oraz wyznaczenie funkcji dla licznika. W tym kroku należy utworzyć tablicę prawdy dla dekodera, a następnie na jej podstawie wyznaczyć funkcje wzbudzeń dla licznika. Wejściami układu dekodera są funkcje Z oraz Q (wyjście rejestru), natomiast wyjście to funkcja wzbudzeń dla licznika.

5. Implementacja mikroprogramowanego układu sterującego. Ostatni etap to implementacja układu. W przypadku matryc programowalnych FPGA pamięć sterownika (CM) może zostać zaimplementowana na dwa sposoby. Pierwsza możliwość to realizacja z wykorzystaniem dedykowanych bloków pamięci matryc FPGA. Takie rozwiązanie pozwala znacznie zmniejszyć ilość wykorzystanych bloków logicznych. Drugi sposób to implementacja pamięci z wykorzystaniem bloków logicznych FPGA. Opcja ta jest stosowana zazwyczaj tylko wtedy, gdy rozmiar pamięci sterownika przekracza rozmiar dostępnego miejsca w pamięciach dedykowanych [6, 7].

6. Przykład syntezy mikroprogramowanego układu sterującego z dekoderni adresów

Przedstawiona metoda syntezy mikroprogramowanych układów sterujących z wykorzystaniem dekodera adresów zostanie zilustrowana przykładem. W tym celu wykorzystany zostanie hipotetyczny układ sterowania U_1 , opisany za pomocą sieci działań przedstawionej na rys. 4.



Rys. 4. Przykładowa sieć działań układu sterującego U_1
Fig. 4. Initial Flow-Chart of CMCU U_1

Aby metoda wykorzystująca dekodery adresów była efektywniejsza od tradycyjnej metody współdzielenia kodów, warunek (1) nie może być spełniony. W pokazanym przykładzie sieć działań zawiera dwa bloki warunkowe (x_1 oraz x_2) oraz osiem bloków operacyjnych ($b_1, \dots, b_8$). Istnieją trzy łańcuchy bloków operacyjnych $C = \langle a_1, a_2, a_3 \rangle$, gdzie $a_1 = \langle b_0 \rangle$, $a_2 = \langle b_0, b_1 \rangle$, $a_3 = \langle b_2, \dots, b_7 \rangle$. Stąd wynika, że rejestr RG może zostać zakodowany z wykorzystaniem dwóch zmiennych ($R_1 = \lceil \log_2 3 \rceil = 2$). Ilość bloków w najdłuższym łańcuchu (a_3) jest równa 5, dlatego też rozmiar kodu bloku operacyjnego wynosić będzie 3 ($R_2 = \lceil \log_2 5 \rceil = 3$). Rozmiar adresu mikroinstrukcji jest równy 3 (osiem mikroinstrukcji, stąd $R_3 = \lceil \log_2 8 \rceil = 3$), więc warunek (1) nie jest spełniony. Co więcej, $R_1 + R_2$ jest większe od R_3 , aż o 2 jednostki, co oznacza aż czterokrotne zwiększenie rozmiaru pamięci. Rozwiązaniem tego problemu może być zastosowanie metody z dekoderni adresów.

Pierwszym krokiem, jaki należy wykonać podczas syntezy mikroprogramowanego układu sterującego z dekoderni adresów jest utworzenie łańcuchów bloków operacyjnych oraz wyznaczenie zawartości pamięci sterownika.

Jak wspomniano wcześniej, w układzie wyróżnić można 3 łańcuchy bloków operacyjnych. W tabeli 1 przedstawiono zawartość pamięci sterownika.

Tab. 1. Zawartość pamięci mikroprogramowanego układu sterującego U_1
Tab. 1. Content of Control Memory of CMCU U_1

Adres	Wartość	Blok	Adres	Wartość	Blok
000	y_1	b_0	000	y_1	b_4
001	y_5	b_1	001	y_5	b_5
010	$y_3 y_4$	b_2	010	$y_3 y_4$	b_6
100	$y_3 y_6$	b_3	100	$y_3 y_6$	b_7

Kolejny krok to kodowanie wejść łańcuchów bloków operacyjnych. Łańcuchy α_1 oraz α_3 posiadają jedno wejście (są to odpowiednio bloki b_0 oraz b_2), natomiast łańcuch α_2 ma dwa wejścia: I_1^2 oraz I_2^2 . Zgodnie z (2) $M_4 = 2$, więc $R_4 = 1$. Wynika stąd, że funkcję Z można zakodować z wykorzystaniem jednej zmiennej $Z = \{z_1\}$. Wobec tego wejścia zostaną zakodowane w następujący sposób: $C(I_1^1) = C(I_3^1) = *$, $C(I_2^1) = 0$, $C(I_2^2) = 1$ (symbol „*” oznacza, że nie jest istotne, jaki kod będą miały wejścia łańcuchów, może to być zarówno wartość 0 lub 1).

W następnym etapie należy utworzyć tabelę przejść dla projektowanego układu mikroprogramowanego (tab. 2).

Tab. 2. Tabela przejść mikroprogramowanego układu sterującego U_1
Tab. 2. Table of transactions of CMCU U_1

α_g	$K(\alpha_g)$	x_h	α_t	$K(\alpha_t)$	I_t^j	$C(I_t^j)$	Ψ_h	z_h	h
α_1	00	$\overline{x_1}$	α_1	00	I_1^1	*	--	--	1
		x_1	α_2	01	I_2^1	0	D_2	--	2
α_2	01	x_2	α_2	01	I_2^2	1	D_2	z_1	3
		$\overline{x_2}$	α_3	10	I_3^1	*	D_1	--	4
α_3	10	--	--	--	--	--	--	--	5

Z powyższej tabeli można wyznaczyć funkcje Ψ oraz Z . Dla przykładu:

$$D_2 = \overline{Q_1} * \overline{Q_2} * x_1 + \overline{Q_1} * Q_2 * x_2 \quad (3)$$

$$z_1 = \overline{Q_1} * Q_2 * x_2 \quad (4)$$

Kolejny krok to utworzenie tabeli prawdy dla dekodera (tab. 3).

Tab. 3. Tabela prawdy dla dekodera adresu układu U_1
Tab. 3. Table of the address decoder of CMCU U_1

α_g	$K(\alpha_g)$	I_g^j	$C(I_g^j)$	$A(I_g^j)$	Φ_m	h
α_1	00	I_1^1	*	000	--	1
α_2	01	I_2^1	0	001	C_3	3
		I_2^2	1	010	C_2	4
α_3	10	I_3^1	*	011	$C_2 C_3$	5

Na podstawie tabeli 3 należy wyznaczyć równania dla licznika, przykładowo:

$$C_2 = \overline{Q_1} * Q_2 * z_1 + Q_1 * \overline{Q_2} \quad (5)$$

Ostatni etap to implementacja mikroprogramowanego układu sterującego. Przykładowo dla macierzy programowalnych blok kombinacyjny, rejestr, licznik oraz dekodery adresów są realizowane z wykorzystaniem bloków logicznych. Pamięć sterownika zazwyczaj jest implementowana w dedykowanych blokach pamięci układu FPGA. Szczegółowe informacje na temat implementacji mikroprogramowanych układów sterujących w macierzach programowalnych można znaleźć w literaturze [3, 6, 7].

7. Podsumowanie

W referacie przedstawiono metodę syntezy mikroprogramowanych układów sterujących ze współdzieleniem kodów z wykorzystaniem konwertera adresów. Proponowana metoda umożliwia wykorzystanie współdzielenia kodów, gdy rozmiar adresów generowanych przez licznik oraz rejestr jest większy niż rozmiar adresu mikroinstrukcji.

Wstępne badania przeprowadzone przez autora wykazały skuteczność zaproponowanego rozwiązania. W porównaniu do tradycyjnej metody projektowania mikroprogramowanych układów sterujących, zastosowanie dekodera adresów pozwoliło zmniejszyć ilość wykorzystanych bloków logicznych w docelowych układach o około 10-15%.

Pracę wykonano w ramach projektu badawczego finansowanego ze środków Zintegrowanego Programu Operacyjnego Rozwoju Regionalnego (Działanie 2.6: Regionalne strategie innowacyjne i transfer wiedzy) z udziałem Europejskiego Funduszu Społecznego.

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 macierzach programowalnych, KNWS'05, Oficyna Wydawnicza UZ, Zielona Góra, 2005.