

Remigiusz WIŚNIEWSKI¹, Alexander BARKALOV¹, Wolfgang A. HALANG²

¹UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

²FERNUNIVERSITÄT HAGEN, LEHRGEBIET INFORMATIONSTECHNIK

Projektowanie sterowników mikroprogramowanych z wykorzystaniem bloków pamięci układów programowalnych

Dr inż. Remigiusz WIŚNIEWSKI

Dr 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. dr hab. inż. 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

Prof. dr hab. inż. Wolfgang A. HALANG

Profesor Wolfgang A. Halang tytuły doktora uzyskał w dwóch specjalnościach: matematyka (Uniwersytet w Bochum) oraz informatyka (Uniwersytet w Dortmundzie). Pełnił funkcję dziekana wydziału informatyki w Uniwersytecie w Groningen. Obecnie jest dziekanem wydziału informatyki w Uniwersytecie w Hagen. Jego zainteresowania badawcze koncentrują się wokół zagadnień związanych z bezpieczeństwem systemów czasu rzeczywistego.



e-mail: wolfgang.halang@FernUni-Hagen.de

Streszczenie

W artykule zaprezentowana zostanie nowa metoda projektowania sterowników mikroprogramowanych. W proponowanym rozwiązaniu zarówno moduł pamięci, jak i część adresująca sterownika realizowane są z wykorzystaniem dedykowanych bloków pamięci układów programowalnych. Dzięki temu układ sterujący może zostać w łatwy sposób zweryfikowany pod kątem bezpieczeństwa. Ponadto zredukowana zostaje liczba wykorzystanych elementów logicznych, które są niezbędne do implementacji układu sterującego w matrycach programowalnych. Trzecią istotną korzyścią proponowanej metody jest możliwość zastosowania częściowej rekonfiguracji zarówno części zarządzającej jak i modułu pamięci sterownika. Idea metody zostanie zilustrowana przykładem. Pokazane zostaną wszystkie kroki, które są niezbędne do realizacji układu z wykorzystaniem proponowanej metody.

Słowa kluczowe: sterownik mikroprogramowany (mikroprogramowany układ sterujący), programowalny układ FPGA.

Design of microprogrammed controllers with dedicated memory blocks

Abstract

A compositional microprogram control unit (also called a microprogrammed controller) is a multi-level device whose control unit consists of two main units. The first one is responsible for addressing microinstructions that are kept in a control memory. It is a simple finite-state machine. The role of the second unit is to hold and generate adequate microinstructions. Such a solution permits to minimize the number of logic elements required to implement the control unit. Therefore, wider areas of the target device can be accessed by other modules of the designed system. The control memory can be implemented using either logic elements or dedicated memory blocks of a chip. In this paper a new design method of microprogrammed controllers is proposed. Its idea is to implement the addressing part of microprogrammed controllers with memories. This kind of solutions has three main advantages. First of all, such designs can easily be verified thanks to the regular structure of memories, which fosters the inherent safety of entire control units. Moreover, in such a case the design is implemented with a System-on-Programmable-Chip (SoPC), the additional resources available there in form of programmable devices may be used. It is possible to implement the addressing part of a control unit with dedicated memory blocks of an SoPC, which results in a reduction of

logic elements (especially look-up tables) required. Finally, the concept allows application of partial reconfiguration of an SoPC. Thus, the functionality of the whole controller can be easily and quickly modified. The proposed method is illustrated by an example. All steps required in order to design and prototype microprogrammed controllers based on the presented concept are shown in detail.

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

1. Wstęp

Jednostka sterująca jest ważną częścią projektowanego systemu cyfrowego [1, 2, 3, 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, 6]. Coraz częściej spotykanym rozwiązaniem jest sterownik mikroprogramowany (zwany także mikroprogramowanym układem sterującym), w którym zastosowano dekompozycję strukturalną jednostki sterującej na część zarządzającą (adresującą) oraz pamięć, w której przechowywane są mikroinstrukcje kontrolera [2, 7, 8].

W dotychczasowych pracach dotyczących sterowników mikroprogramowanych, pamięć układu sterującego implementowana była w dedykowanych blokach pamięci matryc FPGA, natomiast część adresująca mikrokontrolera realizowano z wykorzystaniem logiki programowalnej systemu cyfrowego [2, 8, 9]. Takie podejście ma dwie istotne zalety. Po pierwsze pozwala zredukować liczbę wykorzystanych elementów logicznych poprzez zastosowanie dedykowanych bloków pamięci. Drugą korzyść to możliwość zastosowania częściowej rekonfiguracji sterowników mikroprogramowanych (wymieniany jest tylko fragment pamięci, pozostała część sterownika pozostaje bez zmian, szczegółowy opis częściowej rekonfiguracji układów mikroprogramowanych można znaleźć w [10]).

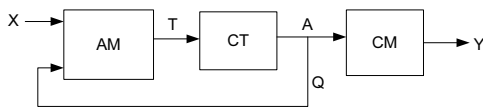
W artykule zaprezentowana zostanie nowa metoda projektowania sterowników mikroprogramowanych. W proponowanym rozwiązaniu zarówno moduł pamięci, jak i część adresująca sterownika realizowane będą z wykorzystaniem dedykowanych bloków pamięci matryc FPGA [11, 12, 13]. Dzięki temu układ sterujący może zostać w łatwy sposób zweryfikowany pod kątem bezpieczeństwa [14]. Wiąże się z regularną strukturą pamięci, która w przedstawionym rozwiązaniu jest zastosowana także do realizacji części zarządzającej sterownika [15]. Drugą zaletą proponowanej metody jest redukcja wykorzystanych elementów logicznych, które są niezbędne do implementacji układu sterującego w matrycach programowalnych. Realizacja części zarządzającej z wykorzystaniem bloków pamięci pozwala zmniejszyć liczbę wykorzystanych elementów LUT (Look-Up Tables) w porównaniu do rozwiązań tradycyjnych. Trzecią istotną korzyścią proponowanej metody jest możliwość zastosowania częściowej

wej rekonfiguracji zarówno części zarządzającej jak i modułu pamięci sterownika. Dzięki temu w bardzo szybki i prosty sposób zmodyfikowana może zostać zarówno funkcjonalność jak i zawartość instrukcji generowanych przez sterownik.

W referacie przedstawiono tylko najważniejsze definicje dotyczące sterowników mikroprogramowanych. Szczegóły dotyczące projektowania mikroprogramowanych jednostek sterujących można znaleźć w literaturze [2, 8] oraz licznych artykułach [9, 10, 12, 16, 17].

2. Mikroprogramowany układ sterujący

Jednostka sterująca może zostać zrealizowana jako mikroprogramowany układ sterujący [2, 4, 7, 9]. Podstawową cechą takiego systemu jest podział mikrokontrolera 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 poprzez rys. 1 moduł adresujący (ang. *Addressing Module*, AM) odpowiedzialny jest za wyznaczenie funkcji wzbudzeń T dla licznika CT. W standardowych rozwiązaniach, blok ten realizowany jest jako układ kombinacyjny, z wykorzystaniem elementów logicznych matryc FPGA [9, 10, 17].

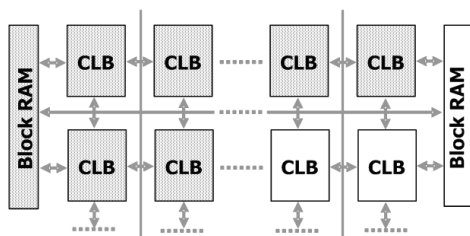
Licznik (ang. *Counter*, CT) oraz pamięć sterownika (ang. *Control Memory*, CM) tworzą blok przechowujący oraz generujący mikroinstrukcje. W układzie dodatkowo zastosowano kodowanie mikroinstrukcji, dzięki czemu kod stanu Q kontrolera stanowi część adresu A mikroinstrukcji:

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

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 możliwość implementacji modułu pamięci sterownika w dedykowanych blokach pamięci matrycy FPGA. Pozostałe bloki systemu (moduł AM oraz CT) realizowane są z wykorzystaniem elementów logicznych LUT układu programowalnego (rys. 2).



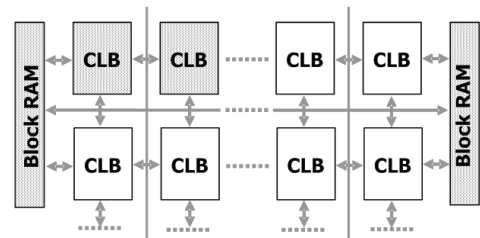
Rys. 2. Standardowa implementacja sterownika w układzie FPGA
Fig. 2. Typical realization of a microprogrammed controller with FPGA

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 dedy-

kowanych bloków pamięci matrycy FPGA, pozostałe elementy pozostają bez zmian. Metoda częściowej rekonfiguracji sterowników mikroprogramowanych implementowanych z wykorzystaniem układów programowalnych została bardzo szczegółowo przedstawiona w [10].

3. Idea proponowanej metody

Jak wspomniano w poprzednim rozdziale, układ przedstawiony na rys. 1 jest realizowany poprzez implementację modułu pamięci w dedykowanych blokach pamięci, natomiast moduł adresujący oraz licznik realizowany jest z wykorzystaniem elementów logicznych matrycy FPGA [18]. W niniejszym artykule zaproponowano nowatorski sposób realizacji sterownika mikroprogramowanego w strukturach FPGA. Proponujemy, aby zarówno moduł pamięci jak i blok adresujący były implementowane z wykorzystaniem dedykowanych bloków pamięci. W ten sposób jedynie licznik jest realizowany z wykorzystaniem elementów logicznych układu FPGA (rys. 3).



Rys. 3. Implementacja sterownika z wykorzystaniem proponowanej metody
Fig. 3. Realization of the microprogrammed controller with the proposed method

W przedstawionym rozwiązaniu wyróżnić można trzy podstawowe korzyści w porównaniu do klasycznej implementacji układów mikroprogramowanych:

- **Redukcja elementów logicznych, jakie są niezbędne do realizacji sterownika w strukturach programowalnych.** Dzięki implementacji bloku AM z wykorzystaniem dedykowanych bloków pamięci matrycy FPGA, znacznie zmniejszona zostaje liczba wykorzystanych elementów LUT (ang. Look-Up Tables [4, 9, 18]). W ten sposób większy obszar układu programowalnego może zostać zagospodarowany przez inne elementy projektowanego systemu. Oczywiście należy tutaj także pamiętać, że taki sposób realizacji mikrokontrolera wymusza zastosowanie co najmniej dwóch dedykowanych bloków pamięci układu FPGA (jeden jest wykorzystany przez moduł adresujący, drugi przez moduł pamięci sterownika).
- **Weryfikacja układu pod kątem bezpieczeństwa.** Implementacja bloku AM z wykorzystaniem dedykowanych bloków pamięci matrycy FPGA oznacza, że większa część sterownika realizowana jest z wykorzystaniem pamięci. Taki sposób realizacji pozwala w łatwy sposób zweryfikować układ pod kątem bezpieczeństwa. Wynika to z regularnej struktury pamięci. Szczegółowe badania dotyczące bezpieczeństwa układów sterujących implementowanych z wykorzystaniem pamięci znaleźć można w [14, 15].
- **Możliwość zastosowania częściowej rekonfiguracji modułu adresującego.** Proponowane rozwiązanie pozwala wykorzystać mechanizm częściowej rekonfiguracji nie tylko w przypadku modułu pamięci, ale także do szybkiej wymiany zawartości modułu adresującego [10, 19]. W praktyce oznacza to możliwość zmiany funkcjonalności całego sterownika, a jedynym ograniczeniem dla projektanta jest rozmiar danych generowanych przez licznik (moduł licznika jest implementowany z wykorzystaniem

elementów logicznych matryc FPGA i nie może być zmodyfikowany z wykorzystaniem proponowanej idei częściowej rekonfiguracji). Należy tu wyraźnie zaznaczyć, że w przypadku tak zrealizowanego sterownika, w bardzo szybki i łatwy sposób zmodyfikowana może zostać zarówno zawartość pamięci układu mikroprogramowanego (moduł pamięci), jak i funkcjonalność sterownika (moduł adresujący).

W następnym rozdziale przedstawione zostaną poszczególne kroki projektowania sterownika mikroprogramowanego, jakie są niezbędne do późniejszej implementacji mikrokontrolera z wykorzystaniem dedykowanych bloków pamięci matryc FPGA.

4. Realizacja układu mikroprogramowanego z wykorzystaniem proponowanej metody

Realizacja mikroprogramowanego układu sterującego przebiega w następujących etapach:

1. **Utworzenie 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, 8, 9, 10, 12, 17]. 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 [17].
3. **Utworzenie tabeli przejść mikroprogramowanego układu sterującego.** 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 . Tabela przejść określa zawartość modułu adresującego, który jest implementowany z wykorzystaniem dedykowanych bloków pamięci matryc FPGA.
4. **Implementacja układu z wykorzystaniem matryc programowalnych FPGA.** Zarówno pamięć sterownika jak i moduł adresujący są implementowane z wykorzystaniem dedykowanych bloków pamięci matryc FPGA. Licznik jest realizowany poprzez elementy logiczne układu programowalnego. Ze względu na strukturę dedykowanych bloków pamięci matryc FPGA, moduły AM oraz CM są implementowane jako jednostki synchroniczne.

5. Wnioski

W referacie zaprezentowana została nowatorska idea projektowania oraz implementacji sterowników mikroprogramowanych. Zaproponowane rozwiązanie bazuje na maksymalnym wykorzystaniu dedykowanych bloków pamięci do realizacji sterownika. Takie podejście pozwala znacznie zmniejszyć liczbę

wykorzystanych elementów logicznych matryc programowalnych. Ponadto zastosowanie pamięci zwiększa bezpieczeństwo całego układu, ze względu na regularną strukturę i łatwość weryfikacji. Najważniejszą jednakże zaletą proponowanej metody jest możliwość częściowej rekonfiguracji zarówno modułu adresującego, jak i modułu pamięci mikrokontrolera. Dzięki temu w bardzo prosty i szybki sposób może zostać zmodyfikowana nie tylko zawartość generowanych mikroinstrukcji, ale także funkcjonalność sterownika.

6. Literatura

- [1] E. Sentovich, K.J. Singh, C.W. Moon, H. Savoj, R.K. Brayton, A.L. Sangiovanni-Vincentelli: Sequential circuit design using synthesis and optimization, ICCD '92: Proceedings of the 1991 IEEE International, 1992, pp. 328-333.
- [2] S. Baranov: Logic Synthesis for Control Automata, Kluwer Academic Publishers, 1994.
- [3] V. Solovjev: Design of the Functional Units of Digital Systems Using Programmable Logic Devices, Bestprint, Minsk, 1996.
- [4] T. Łuba (Praca zbiorowa pod redakcją prof. Tadeusza Łuby): Synteza układów cyfrowych, WKŁ, Warszawa, 2003.
- [5] D. Kania: Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL, Zeszyty Naukowe Politechniki Śląskiej, Gliwice, 2004.
- [6] G. DeMicheli: Synthesis and Optimization of Digital Circuits, McGraw Hill, New York, 1994.
- [7] M. Molski: Modułowe i mikroprogramowalne układy cyfrowe, WKŁ, Warszawa, 1986.
- [8] A. Barkalov, A.V. Palagin: Synthesis of Microprogram Control Units, IC NAS of Ukraine, Kiev, Ukraine, 1997.
- [9] A. Barkalov: Synteza jednostek sterujących w strukturach programowalnych, KNWS'05, Oficyna Wydawnicza UZ, Zielona Góra, 2005.
- [10] R. Wiśniewski: Częściowa rekonfiguracja mikroprogramowanych układów sterujących implementowanych z wykorzystaniem struktur FPGA, OWD 2005, Archiwum Konferencji PTETiS, Wisła, 2005.
- [11] Xilinx: Using Block SelectRAM+ Memory in Spartan-II FPGAs, www.xilinx.com/bvdocs/appnotes/xapp130.pdf, 2000.
- [12] R. Wiśniewski: Projektowanie układów mikroprogramowanych z wykorzystaniem wbudowanych bloków pamięci w matrycach programowalnych, KNWS'05, Oficyna Wydawnicza UZ, Zielona Góra, 2005.
- [13] Xilinx: Using Block RAM in Spartan-3 Generation FPGAs, www.xilinx.com/bvdocs/appnotes/xapp463.pdf, 2005.
- [14] W.A. Halang, M. Adamski: A programmable electronic system for safety related control applications, Advances in safety and reliability, Proceedings of the International Conference - ESREL '97, Vol. 1, Pergamon, Oxford, 1997, pp. 349-355.
- [15] W.A. Halang, B.J. Krämer: Safety assurance in process control, IEEE Softw., Vol 11(1), 1994, pp. 61-67.
- [16] C.A. Papachristou: A scheme for implementing microprogram addressing with programmable logic arrays, Digital Processes Vol. 5 (No. 3-4), 1979, pp. 235-256.
- [17] R. Wiśniewski, A. Barkalov, L. Titarenko: Optimization of address circuit of CMCU, EWDTW '06. Kharkov, 2006, pp. 167-170.
- [18] C. Maxfield: The Design Warrior's Guide to FPGAs, Academic Press, Inc., Orlando, FL, USA, 2004.
- [19] Xilinx: Two flows for partial reconfiguration, <http://direct.xilinx.com/bvdocs/appnotes/xapp290.pdf>, 2004.