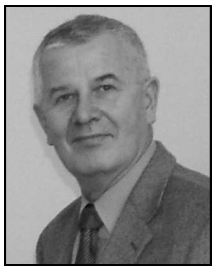
Marian ADAMSKI, Alexander BARKALOV, **Remigiusz WIŚNIEWSKI**, Jakub LIPIŃSKI
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI, UL. LICEALNA 9, 65-417 ZIELONA GÓRA

Reduction of the Memory Size in the Microprogrammed Controllers

prof. dr hab. inż. Marian Adamski



Profesor zwyczajny, zatrudniony w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Zainteresowania badawcze obejmują projektowanie systemów cyfrowych realizowanych w postaci mikrosystemów cyfrowych oraz formalnych metod programowania sterowników logicznych.

e-mail: M.Adamski@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 w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego.

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

dr inż. Remigiusz Wiśniewski

Absolwent Uniwersytetu Zielonogórskiego, pracę doktorską obronił w 2008 roku. W latach 2000-2001 dwukrotnie odbył przemysłową praktykę studencką w firmie Aldec Inc. w Stanach Zjednoczonych. Aktualnie pracuje jako adiunkt (Uniwersytet Zielonogórski). Zainteresowania badawcze obejmują zagadnienia z zakresu teorii grafów i hipergrafów, bezpieczeństwa danych i kryptologii oraz metodologii projektowania i implementacji systemów cyfrowych.

e-mail: R.Wisniewski@iie.uz.zgora.pl

mgr inż. Jakub Lipiński

Absolwent Uniwersytetu Zielonogórskiego, pracę magisterską obronił w 2009 roku. Jest słuchaczem studiów doktoranckich, specjalność informatyka. Aktualnie pracuje w Szpitalu Wojewódzkim SPZOZ w Zielonej Górze. Zainteresowania badawcze obejmują zagadnienia z zakresu teorii grafów i hipergrafów.

e-mail: J.Lipinski@weit.uz.zgora.pl

Abstract

The method of reduction of the control memory size in the microprogrammed controllers is proposed in the article. The idea is based on the hypergraph theory. The concurrent microoperations are encoded together thus the total volume of the memory is reduced. In order to receive the proper microinstruction, an additional module – microinstruction decoder is also prepared. The idea of the proposed method is illustrated by an example. Moreover, the result of performed experimental investigations is presented, as well.

Keywords: hypergraphs, microprogrammed controllers, reduction, memory, microinstruction decoder.

Redukcja pojemności pamięci w sterownikach mikroprogramowanych

Streszczenie

W artykule zaproponowano metodę redukcji pojemności pamięci sterowników mikroprogramowanych. Metoda bazuje na teorii hipergrafów. Mikrooperacje parami kompatybilne są kodowane wspólnie, dzięki czemu redukcji ulega całkowita pojemność pamięci sterownika mikroprogramowanego. Do struktury układu wprowadzono dodatkowy moduł, dekodera mikroinstrukcji. Jednostka ta jest odpowiedzialna za odkodowanie pierwotnych danych. Idea proponowanej metody zilustrowano przykładem. Ponadto, przeprowadzono także badania eksperymentalne, których celem była weryfikacja skuteczności proponowanej metody. Wyniki badań pokazują, że pierwotna pamięć sterownika jest redukowana średnio o 21%.

Słowa kluczowe: hipergrafy, sterowniki mikroprogramowane, redukcja, pamięć, konwerter adresów.

1. Introduction

A control unit (CU) is important part of a digital system [1,2,3,4,5,6]. Usually, the control unit is realized as a finite state machine (FSM) [4,6,7,8]. However, in the case of the linear flowchart, the microprogrammed controller may require less amount of hardware than control unit based on the traditional FSM model [9,10,11]. In case of microprogrammed controllers, the control unit is decomposed into two main parts. The first one addresses microinstructions [7,9], while the second is in response of holding and generating the proper microinstruction [10,12].

Such a solution leads to the reduction of the number of logic elements that are required for implementation of the controller [12]. Typically, the control memory is implemented as a ROM or RAM memory. Thus, wider areas of the destination device can be used for other modules of the prototyped system [1,4,5,7,10,13].

Most of controllers (especially realized as a Complex Instruction Set Computers, CISC) have a long microinstruction width what influences on the memory size [3,5]. Such a situation causes serious problems in the prototyping process. In case of System-On-Programmable-Chip (SoPC), the memory can be implemented with dedicated memory blocks of the Field Programmable Gate Arrays (FPGA). However, if the microinstruction length exceeds the total length of the dedicated memory block of an FPGA, the memory has to be decomposed. On the other hand, in case of controllers implemented as a System-On-Chip (SoC), the memory is designed as an independent module. This means that each additional bit in the microinstruction width increases the total size of the memory and increases the cost of the whole device.

In the paper we propose the method of the control memory reduction. The idea is based on the reduction of the microinstruction length by encoding the concurrent microoperations together. To achieve it, the hypergraph theory is applied [11,14]. Moreover, the particular stages of the reduction process also are performed with hypergraphs. Finally, the initial memory is reduced, while the proper microinstructions are decoded by an additional block of microinstruction decoder.

2. Idea of the proposed method

To reduce the memory size, microinstructions that are pairwise compatible (can be executed concurrently) will be encoded together. Such an idea permits to reduce the total memory size. However, an additional block, Microinstruction Decoder (MD) is also required. Figure 1 illustrates the idea of the proposed method.

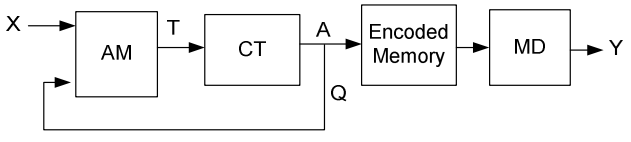


Fig. 1. An idea of the proposed method
Rys. 1. Idea proponowanej metody

The reduction method bases on the hypergraph theory, initially presented in [11,14,15]. The reduction idea was enhanced and adapted to the microprogrammed controllers. The whole process can be divided into the following steps:

1. Formation of the set of compatibility classes.
2. Determination of the weight (cost) of each compatibility class.
3. Calculation of the hypergraph dual to the hypergraph H_C.
4. Determination of the minimum vertex covering (transversal) of hypergraph H_D.
5. Calculation of the total cost of each minimum covering and reduction of redundant microoperations.
6. Encoding compatibility classes which realize minimal transver sal.
7. Determination of a new content of memory with encoded com patibility classes.
8. Formation of the module of Microinstruction Decoder. In the final step the module MD is formed.

Finally, the whole system can be designed. Depending on the designers demands, proper modules can be implemented in various ways [3,4,12,16,17]. For example the addressing module, the counter and the microinstruction decoder may be realized with Look-Up Tables (LUTs), while Encoded Memory can be implemented with dedicated memory blocks of a destination reprogrammable device [5,14].

3. The results of experiments

The effectiveness of the discussed method has been verified experimentally. The library of test modules consists of over 50 benchmarks. All of examined memories were real memory systems of logic controllers and they were taken from specifications and benchmarks presented in [7,9,10,11,12,14,18].

Table 1 shows the results of experiments, where representative benchmarks (memories) are presented.

Tab. 1. Results of experiments
Tab. 1. Wyniki badań eksperymentalnych

Benchmark	Size of the initial memory [bits]	Size of the reduced memory [bits]	Density of the memory	Reduction of the memory [%]
Test022	264	154	19,31%	42%
Test021	176	112	20,45%	36%
TestMK_06	234	162	28,63%	31%
TestMK_10	180	135	27,22%	25%
Test016	25	20	28,00%	20%
Test032	168	140	41,66%	17%
Test012	60	55	71,66%	8%
Average	**72,93**	**57,58**	**35,66%**	**21%**

From above tables we can see that application of the proposed method permits to reduce the volume of the memory on average by 21%. All of the examined memories were reduced (the lowest reduction was equal to 8%, while the highest – 42%). It means, that the size of reduced memory in microprogrammed controllers is always smaller than the initial one.

It is worth to notice, that effectiveness of the reduction process strongly depends on the density of the initial memory. Results of experiments have shown that reduction is especially high in case of density lower than 30% of the initial memory. For example the memory of Test022 can be reduced over than 40%. On the other hand, there are controllers which memories are hard to reduce with higher density, and the reduction may be less than 10%.

4. Conclusions

The method of reduction of microprogrammed controllers memory was presented in the paper. The idea was based on the hypergraph theory. An initial memory is represented by a hypergraph. Further computations lead to the reduced memory, where microoperations are encoded. The results of experiments have shown, that effectiveness of the proposed method strongly depends on the density of the initial memory.

5. References

[1] De Micheli G., Synthesis and Optimization of Digital Circuits, McGraw-Hill, New York, NY, (1994)
[2] Doligalski M., Adamski M., Hierarchical configurable Petri net modeling in VHDL, International Journal of Electronics and Telecommunications, 58 (4), (2012), pp. 397-402
[3] Gajski D., Principles of Digital Design, Prentice Hall, Upper Saddle River, NJ, (1996)
[4] Kania D., The Logic Synthesis for the PAL-based Complex Programmable Logic Devices, Lecture Notes of the Silesian University of Technology (in Polish), Gliwice, (2004)
[5] Maxfield C., The Design Warrior's Guide to FPGAs, Academic Press, Inc., Orlando, FL, (2004)
[6] Sentovich, E.M., Sequential Circuit Synthesis at the Gate Level, Ph.D. thesis. Chair-Robert K. Brayton, (1993)
[7] Baranov S.I., Logic Synthesis for Control Automata, Kluwer Academic Publishers, Boston, MA, (1994)
[8] Łuba T., Synthesis of Logic Devices, Warsaw University of Technology Press (in Polish), Warsaw, (2005)
[9] Adamski M., and Barkalov A., Architectural and Sequential Synthesis of Digital Devices, University of Zielona Góra Press, Zielona Góra, (2006)
[10] Barkalov A., and Titarenko L., Logic synthesis for FSM-based control units, Lecture Notes in Electrical Engineering, Vol. 53. Springer-Verlag, Berlin, (2009)
[11] Wiśniewska M., Wiśniewski R., and Adamski M., Reduction of the microinstruction lenght in the designing process of microprogrammed controllers, Electrical Review, 85 (7) (2009), 203–206
[12] Wiśniewski R., Synthesis of compositional microprogram control units for programmable devices, Lecture Notes in Control and Computer Science, Vol. 14. University of Zielona Góra Press, Zielona Góra, 2009
[13] Berge C., Graphs and Hypergraph, North-Hols.r Mathematical Library, Amsterdam (1976)
[14] Wiśniewska M., Application of hypergraphs to the decomposition of the discrete-systems, PhD thesis, University of Zielona Góra, (2011)
[15] Adamski M., Wiśniewska M., Wiśniewski R., Stefanowicz Ł., Application of hypergraphs to the reduction of the memory size in the Microprogrammed Controllers with Address Converter, Electrical Review, 8 (2012), 134–136
[16] Grobelna I., Formal verification of embedded logic controller specification with computer deduction in temporal logic, Electrical Review, 12a (2011), 40-43
[17] Milik A., Hrynkiewicz E., Synthesis and implementation of reconfigurable PLC on FPGA platform, International Journal of Electronics and Telecommunications, 58 (1) (2012), 85–94
[18] Kołopieńczyk M., Application of adress converter for decreasing memory size of CMCU with code sharing, Lecture Notes in Control and Computer Science, Vol. 12. University of Zielona Góra Press, Zielona Góra, (2008)