DOI: 10.35598/mcfpga.2023.009

“Microprogramming” Course in the Knowledge Field 12 “Information Technology”

Mykhailo Petryshyn
ORCID 0000-0001-6319-3768
dept. of Computer Science adh Information Systems
Precarpathian National University
Ivano-Frankivsk, Ukraine
m.l.petryshyn@pnu.edu.ua

Lubomyr Petryshyn
ORCID 0000-0003-4168-3891
dept. of Enterprise Management
AGH University of Krakow
Krakow, Poland
lpetr@agh.edu.pl

***Abstract*—Microprogramming is used to implement the control logic of a computer’s central processing unit at a low level of programming. The microprogram code specifies the sequence of the processor elementary operations execution, ensures the order of complex instruction sets implementation, and allows the design of high-level programming language commands. The relevance of studying the theory, methods, and techniques of microprogramming in higher education institutions allows students to acquire skills in developing functionally oriented processors structures, their command set, as well as system programming skills. The purpose of studying microprogramming is to gain an understanding of how flexible and a computer’s central processor efficient control is achieved using microcode. This allows microprogram optimization, improving performance and ensuring system functionality. In summary, students will gain an understanding of how hardware and software interact at a low level.**

Keywords—microprogramming, course, processor control, student

I. Introduction

In the development of specialized computing systems, matrix logic circuits and devices with microprogram control and bit-modular organization are widely used. This provides flexibility and versatility, and justifies the technical and economic efficiency of their use [1-5]. The use of such technical equipment allows for the creation of high-speed systems whose architecture takes into account the nature and structure of data, the specifics of the solved problem, and ensures the accuracy of calculations. A clear division of functions and their modular organization allows for easy and efficient implementation of practically any type of logic [4-8]. Devices with bit-modular organization and matrix-type logic circuits are widely used in special-purpose systems [2, 6, 8, 9]. If it is necessary to manufacture a limited quantity of computing systems, it is advisable to use microprogrammed devices [6-8]. However, with a significant planned release of developed systems, it is more economical to use very large-scale integrated circuits. The acquisition by students of theoretical and practical skills in building a computing processor environment and its microprogramming determines the relevance of introducing the academic course "Microprogramming" [9-18]. The purpose of studying this course is to understand the basics of machine languages, low-level languages, and system programming [9, 11, 13].

II. Basic Components of IT

The basic components of information technology (IT) that make up its infrastructure (Fig. 1), and according to which specialists are studied in the knowledge field 12 "Information Technology" are [1-1-4, 6, 8]:

- users of IT;
- environment;
- software;
- technical support.

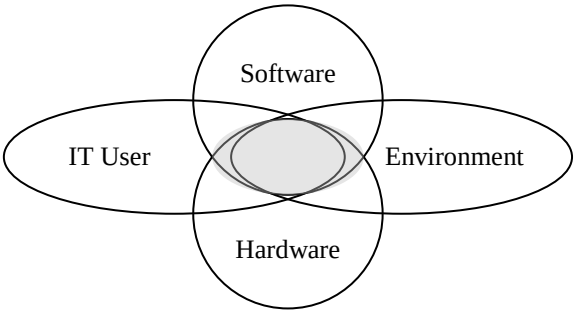

Fig. 1. Infrastructure of the IT basic components.

Students face the most difficulties when studying curses that are adjacent components of information technology (IT). These components are marked in gray on the diagram.

At the same time, it is necessary to define more specifically the components or elements that make up the software (Fig. 2) [3, 4, 6, 9]:

- application software;
- algorithmic support;
- mathematical support;
- operating system;
- system software;
- assemblers;
- machine languages.

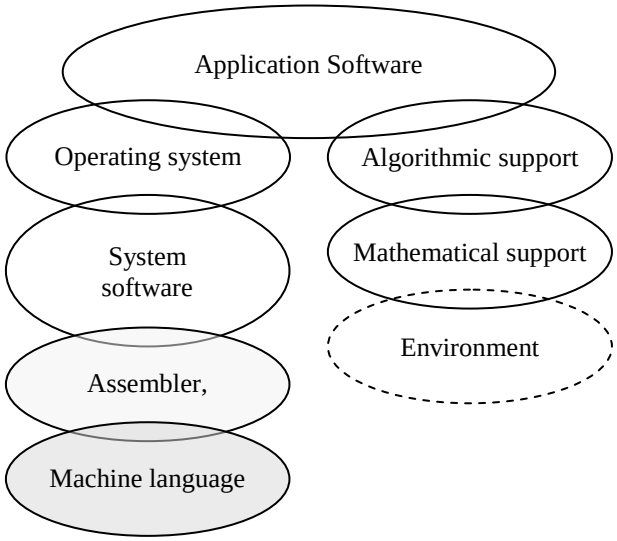

Fig. 2. Hierarchical software components.

As teaching practice shows, students have a "fundamental" gap in understanding the interaction of technical and software components on the microprogramming level (Fig. 3), which creates a psychological barrier to mastering programming languages and understanding the principles of software control and operation of hardware computing tools. Very often, the studying of principles and methods of software control of the computing environment is neglected or omitted, which is based on its dynamic reconfiguration during program execution. Studying high-level languages without a transitional link of microprogrammed control makes it impossible to subjectively understand the basics of programming [10-13, 15-18].

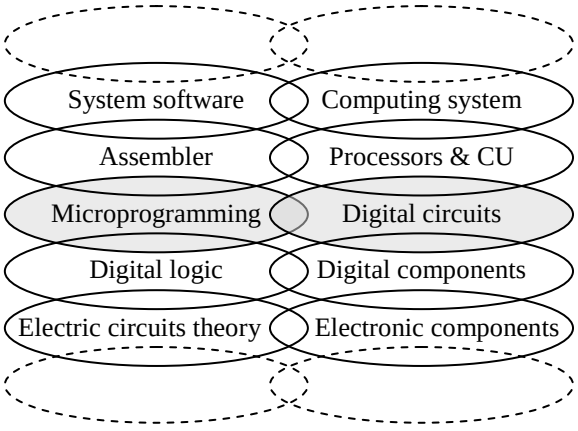

Fig. 3. Software and hardware interaction levels.

III. Implementation of the "Microprogramming" Course

It is proposed to separate the studying of the basics and methods of microprogram control, their integration into low-level languages - assemblers, and the synthesis of high-level programming languages. The relevance of introducing the "Microprogramming" course into the study program is also confirmed by the trends of including similar subjects in the curricula of several leading foreign educational institutions, based on the analysis of materials posted on their internet pages [11-14].

As a result of studying the "Microprogramming" course, a student should understand:

- methods of software control of the computational process course at the hardware level;
- basic principles of microprogramming and the difference between machine code and microprogram;
- methods of building microprograms, including methods of organizing conditional and unconditional program transitions;
- the process of designing microcode, including the process of creating, testing, and supporting microcommands;
- application of microprogramming techniques in designing processors and other computing devices.

Furthermore, a student should know the principles and methods of:

- structural organization and controlled synchronous dynamic reconfiguration of digital computing environments;
- synthesis of microcommands and their organization into program modules;
- organization, construction, and execution of basic components of linear programming and organization of unconditional and conditional transitions;
- synthesis of effective programs at the level of microprogrammed control;
- execution of complex commands at the assembler level, their mathematical and algorithmic structure, as well as their synthesis using microprogramming commands;
- transition to high-level languages and synthesis of commands according to established functional requirements.

A student should be able to:

- analyze the structure of a digital computing environment, evaluate its functional capabilities, and understand the composition of microcommands;
- perform structural analysis and optimization of linear programs, including loops;
- synthesize commands of higher levels of programming languages.

The main courses that support these skills are discrete mathematics, computer circuitry, and algorithmic.

IV. Infrastructure of the "Microprogramming" Course

The infrastructure of the "Microprogramming" course includes the following sections:

- principles and methods of software control of computing environments based on dynamic reconfiguration of their resources;
- synthesis of the structure and microprogram control of a single-bit processor;
- basic addressing methods, microcommand structure, and command decoder synthesis;
- composition of microcommands, their functions, and operation codes;
- methods of organizing, processing, and optimizing linear programs and programs with unconditional and conditional branching;
- synthesis of the structure and microprogram control of a multi-bit processor, specifics of data processing, and command execution;
- methods of organizing the software environment and its processing; state devices;
- methods of synthesizing assembler commands and high-level language commands.

V. Discussion & Conclusions

The feasibility of introducing the "Microprogramming" course is confirmed by the experience of students who have taken it, as well as the practical skills acquired. Recent years' practice shows that teaching the subject significantly improves students' understanding of the software control of computational process, data processing, effective program building methodology, and optimization. It deepens their understanding of the specific usage of high-level programming languages and approaches to programming methods. In general, studying "Microprogramming" course will help students comprehend the principles of computer systems functioning and methods of optimizing their work at the microprogramming level.

Acknowledgment

The publication was funded by a subsidy for the maintenance and development of research potential.

References

[1] W. Stallings, "Computer Organization and Architecture," 11th ed., Pearson, 2021.

[2] J. Ledin, "Modern Computer Architecture and Organization," Packt Publishing, 2020, 560 pages.

[3] M. Murdocca, V. Heuring, "Computer Architecture and Organization," John Wiley & Sons, 2007.

[4] D. Comer, "Essentials of Computer Architecture," 2nd Ed., Chapman and Hall/CRC, 2017, 511 p.

[5] L. Petryshyn, D. Sala, „Architecture of the multiprocessor computerized management systems," In: Zarządzanie przedsiębiorstwem : teoria i praktyka 2014 / Sc red. Piotr Łebkowski, Kraków, Wydawnictwa AGH, 2014, pp. 196–198.

[6] J. D. Dumas, "Computer Architecture: Fundamentals and Principles of Computer Design," 2nd Ed., CRC Press; 2016, 462 p.

[7] L. Petryshyn, "Matematičeskie modeli obsluživaniâ informacionnyh processov v mnogoprocessornyh komp'ûterizovannyh sistemah upravleniâ — Mathematical models of information processes maintenance in multiprocessor computerized control systems," In monography: Information Technology in Selected Areas of Management. / Sci. Ed. Lyubomyr Petryshyn. Published by AGH University of Science and Technology Press. Krakow, 2016. 152 p., pp. 11-32.

[8] P. Whatmough, G.-Y. Wei, D. Brooks, "Deep Learning for Computer Architects (Synthesis Lectures on Computer Architecture)," Morgan & Claypool Publishers, 2017, 124 p.

[9] B. Holdsworth, "Microprocessor Engineering," Elsevier, 2013. 352 p.

[10] A. J. Dos Reis, "RISC-V Assembly," Independently published, 2019, 155 p.

[11] A. Agrawala, T. G. Rauscher, "Foundations of Microprogramming: Architecture, Software, and Applications," New York, Academic Press, 1976, 2021 p.

[12] W. Matthes, "Mikroprogrammierung. Prinzipien, Architekturen, Maschinen," Logos, 2021.

[13] A. J. Dos Reis, "Constructing a Microprogrammed Computer," Second Ed., Independently published, 2022, 189 p.

[14] "Microprogramming," Merriam-Webster.com Dictionary, [Online]. Available: https://www.merriam-webster.com/dictionary/microprogramming. (accessed Mai 08, 2023).

[15] "An Introduction to Microprogramming," IBM Corporation, 1971.

[16] M. J. Flynn, R. F. Rosin, "Microprogramming: An Introduction and a Viewpoint," 1971, pp. 727-731, DOI: 10.1109/T-C.1971.223341.

[17] B. E. Cline, "Microprogramming Concepts and Techniques," Petrocelli Books, 1981, 169 p.

[18] S. S. Husson, "Microprogramming. Principles and Practices," Prentice-Hall, 1970.