Experience of Teaching Computer Organization Course for Multiple Majors and Minors

Man-man PENG¹,*, Qiang WU¹ and Jian-jiang Li²,*

¹College of Information Science and Technology, Hunan University, Changsha, China
²Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China

*Corresponding author.

Keywords: Higher education, Teaching reform, Course teaching, Computer organization course.

Abstract. Computer organization course is essential for undergraduate students of computing disciplines related majors and minors. Adapting the course content to students of different backgrounds and goals is not easy. In this paper, we introduce our experience of teaching the computer organization course for multiple majors and minors ranging from computer architecture to artificial intelligence. A combination of adaptive lectures and flexible lab practices is designed to meet the requirements of different majors and minors. Reactions from students show the effectiveness of our approach.

Introduction

Computer organization course is in the core of the computing disciplines curricula, especially for computer science and engineering related majors and minors [1]. Universities and colleges offering related undergraduate programs have computer organization course or equivalent course in their curricula [2].

To meet the requirements of different disciplines and sub-disciplines, similar computer organization courses with different flavors are provided for students to choose [2, 3, 4]. Some courses emphasize on the hardware aspects with circuits design and implementation on reconfigurable hardware platforms like FPGAs [5, 6, 7, 8], while some others take a software oriented approach exploiting simulators and other software utilities [9, 10, 11, 12, 13].

Through our teaching experience over 10 years, we find it is possible to have an integrated course for multiple majors and minors. The experience we have got is outlined as follows:

- A course design with hardware-software integrated knowledge blocks and lab practice;
- An adaptive approach to lecturing according to students background;
- A set of flexible lab projects to fit different students’ majors and minors.

The paper is organized with this section as an introduction, section 2 for the course background, section 3 for the course design, section 4 for the teaching experience, and section 5 as conclusion.

Course Background

Majors and Minors

Since the higher education reformation in late 1990s, new majors and minors are offered due to rapid progress of computer applications. Now there are totally 8 majors and 21 minors offered in our universities, as listed in Table 1.

Course Management

Computer organization course is included in the curricula of all the 8 majors and 21 minors shown in Table 1. Typically, majors and minors more inclined to software, such as software engineering, artificial intelligence, digital media and data science, have the computer organization course
scheduled earlier in the autumn semester of the second year. On the other hand, majors and minors that requires more training on electronics, like computer architecture, computer network, communication engineering and IoT engineering usually have the computer organization course scheduled in the spring semester of the second year.

Table 1. Majors and minors offered.

<table>
<thead>
<tr>
<th>Major</th>
<th>Minor</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science</td>
<td>Computer Architecture</td>
<td>Hardware</td>
</tr>
<tr>
<td></td>
<td>Software and Computer Theory</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>Computer Application Technology</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>Computer Network</td>
<td>Hardware</td>
</tr>
<tr>
<td>Communication Engineering</td>
<td>Communication System</td>
<td>Hardware</td>
</tr>
<tr>
<td></td>
<td>Signal Processing</td>
<td>Hardware</td>
</tr>
<tr>
<td></td>
<td>Communication Network</td>
<td>Hardware</td>
</tr>
<tr>
<td></td>
<td>Computational Electro-magnetics</td>
<td>Hardware</td>
</tr>
<tr>
<td></td>
<td>Integrated Circuits Design and Test</td>
<td>Hardware</td>
</tr>
<tr>
<td>Software Engineering</td>
<td>Software Engineering Theory</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>Software Engineering Technology</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>Software Service Engineering</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>Domain Software Engineering</td>
<td>Software</td>
</tr>
<tr>
<td>Information Security</td>
<td>Network Security</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>System Security</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>Information Countermeasure</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td>Cryptography and Secrecy Management</td>
<td>Software</td>
</tr>
<tr>
<td>Artificial Intelligence</td>
<td>Intelligent System</td>
<td>Software</td>
</tr>
<tr>
<td>Internet of Things (IoT)</td>
<td>IoT Engineering</td>
<td>Hardware</td>
</tr>
<tr>
<td>Digital Media</td>
<td>Digital Media Technology</td>
<td>Software</td>
</tr>
<tr>
<td>Data Science</td>
<td>Big Data Technology</td>
<td>Software</td>
</tr>
</tbody>
</table>

**Course Design**

**Course Goal**

The goal of the computer organization course is set to let the students know the concepts, principles, and mechanisms behind the interfaces of modern computer systems, and get acquainted with the methods and tools to design hardware and software of modern computers.

**Knowledge Blocks and Lectures**

We select the following knowledge blocks for the computer organization course:

- Review of digital system design
- Instruction set and assembly language
- Arithmetic algorithms
- Processor design
- Memory technology and memory system design
- Input/output technology and interfaces
- Multiprocessing, parallel computing and performance evaluation

The lectures are designed to be flexible so as to adapt to students of different majors and minors.
Basically, if students are from software oriented majors and minors, the lectures are given in a top-down way, starting from how system programmers would see for the underlying hardware. If students are from hardware oriented majors and minors, the lectures take a bottom-up way, making more efforts to explain how the hardware circuits are designed to provide the high level functionalities.

**Practice Tasks**

The practice of the course consists of 4 lab projects:
- Design of a simple ALU (arithmetic logic unit)
- Design of simple ROM and RAM
- Design of a simple CPU
- Design of simple input and output interface

The practice tasks are set to be a little different for students of different majors and minors. For hardware-oriented majors and minors, students are required to complete digital circuits design and verification. For software-oriented majors and minors, students are required to complete software simulators of the components instead.

**Course Experience**

**Overview**

We have conducted the hardware-software-integrated computer organization course for multiple majors and minors in recent years. The responses from students suggest that the adaptive lectures and flexible practice tasks are most helpful.

**Adaptive Lecturing**

With adaptive lecturing, we mean that we give lectures in a way that fitting the knowledge background of the students. We classify the majors and minors as hardware or software oriented. When making lectures, we always try to start from the concepts or principles that students have learned in previous related courses like programming or digital logic design. Obviously, it is a little more difficult to perform adaptive lecturing for students with software background than those with hardware one. So we take a top-down way to explain computer organization concepts and principles to students of software background, leading them from top-level functional view of the underlying hardware to the circuits implementation details in a stepwise manner. Responses from students show the effectiveness of this adaptive lecturing approach, “I see how it works!”, as remarked by some students.

**Flexible Practice Tasks**

Practice tasks are flexible to different majors and minors. Students of hardware-oriented majors and minors are asked to complete circuits design, while students of software-oriented majors and minors are asked to complete simulators of the components. These tasks fit for students’ background well. So students are able to finish practice tasks with the knowledge and skills they have learned before, which give them confidence in learning the computer organization course, making the lab practice more smooth and efficient. During the lab practice, we often hear students say out “I know how to write a RAM simulator with C++ now”, or “I just designed a CPU!”, which make both them and us happy.

**Summary**

This paper introduces the course design of a hardware-software integrated computer organization course for multiple majors and minors. We classify the majors and minors as hardware-oriented and software-oriented, then select a set of knowledge blocks covering both hardware and software
aspects. In lecturing and lab practice, an adaptive approach is performed according to the students’ knowledge background. Responses from students show the effectiveness of the adaptive approach.

In the future, we plan to investigate the interaction between different majors and minors. For example, in practice tasks, let students of different majors and minors reference and verify each others’ work. We think this may strengthen students understanding of the relations between hardware and software of the computer systems.

Acknowledgement

The work is supported by the Education Reform Project of the Hunan University (531111000002) and the Construction Project of the Research-oriented Teaching Demonstration Courses of the University of Science and Technology Beijing (KC2017YJX17).

References


