

Application of Emu8086 in Digital Teaching Mode of Microcomputer Principle and Interface Technology

Yan Zhang^{1,*}, Shengju Sang¹, Yuncai Luan¹, Xiangcai Zhu¹

¹ School of Information Science and Technology, Taishan University, Taian, Shandong 271021, China

* Corresponding author: Yan Zhang (Email: xtzy@163.com)

Abstract: Microcomputer principle and interface technology is a course with strong theory and practice. With the help of digital simulation software Emu8086, it can visualize the difficult theories in the course, enhance the vitality and interest of classroom teaching, break through the time and space limitations of experimental courses, stimulate students' interest in learning, increase their learning opportunities, and cultivate their practical and innovative abilities. Taking traffic light control as an example, this paper shares the application of this digital teaching method in classroom teaching and experimental teaching, and analyzes this simulation teaching method and its teaching effect.

Keywords: Microcomputer Principle, Assembly Language, Simulation Teaching.

1. Current Situation of Curriculum Teaching and Learning

Microcomputer Principle and Interface Technology is the core course and elective course of computer related majors in colleges and universities [1]. Its theoretical difficulty is relatively high, and its practical teaching is relatively difficult to implement. The commonly used teaching method is pure classroom theoretical teaching and experimental box verification experiment [2]. In theory teaching, teachers will be faced with the change process of hardware status such as CPU internal registers, buses, memories, and external devices, as well as the dynamic process of assembly language program operation, which makes it difficult for students to understand these abstract theories and understand the operating mechanism of the internal principle, and attacks students' enthusiasm and initiative in this course. However, in practical teaching, the chips and circuits of the experimental box are mostly fixed, the experimental projects are limited, and most of them are limited to confirmatory experiments. It is difficult for students to participate in the detailed design and expansion design of the experiment. These factors limit students' imagination, resulting in low interest in learning this course [3]. In addition, the number of experimental boxes is insufficient, easy to damage, difficult to maintain, and limited opening time, which makes it difficult for students to preview and review experimental content, consolidate understanding of theoretical knowledge points, and achieve training objectives.

In recent years, Emu8086 software has been introduced into the classroom teaching and practical teaching of microcomputer principle and interface technology, which enables teachers to more vividly demonstrate the working principle, working process and operating effect of microcomputer [4,5]. While stimulating students' enthusiasm for learning, it can help students quickly master complex and difficult knowledge points, establish a rational and perceptual understanding of computer working principle, and enhance innovative thinking in learning. Cultivated the ability to do things independently. This text is student-centered and aims to promote the overall development of students. It shares the teaching cases of the application of digital software Emu8086

to the teaching of microcomputer principles and interface technology, and provides a practical method and solution for cultivating students' ability to operate independently and innovate, and mobilizing their enthusiasm for learning [6,7].

2. Emu8086

2.1. Speech Feature Extraction

Emu8086 is a simulation software designed for 8086 CPU under Windows system. It can simulate every step of the 8086 microprocessor, and can display the changes of registers, memories, stacks, variables and flag bits after each instruction is executed. The simulation system is easy to configure, and the operation interface is simple. It can simulate the actual operations that cannot be completed under the protection mode of the operating system [8].

The syllabus of the course of Microcomputer Principle and Interface Technology is based on 8086, which is the basis of X86 series CPUs and the most easily understood microprocessor by students [9]. However, with the development of computers, new versions of operating systems have started the protection mode, while 8086 does not support the protection mode, which makes it difficult to explain many instructions, logical operations, memory access and other knowledge points in a real system. The introduction of virtual machines has become a mandatory option. The Emu8086 built-in 8086 virtual machine can easily solve this compatibility problem after it is introduced into the classroom teaching of computer principles and interface technology. At the same time, it also makes it easier for students to understand the working principles of computers, learn assembly language more conveniently, participate in classroom teaching more actively and actively, and have more extensive ways of extracurricular learning [10].

Emu8086 integrates text editor, compiler, disassembly, debugger, virtual device and driver, and has a visual 8086 virtual machine environment. Therefore, during the learning process of microcomputer working principle and assembly language programming, Emu8086 can be used to observe the single step and full speed execution of the program in the debugger, such as changes in registers, flag bits, and internal

data of memory. In addition, Emu8086 also provides some virtual devices such as stepping motor, traffic light, LED, temperature sensor, etc. These virtual devices can run in the 8086 virtual machine environment, and are ideal tools for 8086 assembly language programming and learning [11,12].

3. Emu8086 Simulation Teaching Case

The following introduces the application of Emu8086 in the teaching process by taking the teaching case "Traffic Signal Control" as an example. The case includes theoretical basis teaching and experimental verification teaching. Many knowledge points in the course are nested and based on each other, which cannot be completely separated in the teaching process. For example, the early learning of assembly language will lay a solid foundation for understanding and mastering the interface technology later, and the port operation in the interface technology later is involved in the early instruction system. The knowledge points such as "port operation instruction" in Chapter 3 and "assembly language programming" in Chapter 4 lay the foundation for "traffic light design" in Chapter 6 interface technology. Therefore, the teaching design should be based on the principle of gradually deepening from simplicity to difficulty. The teacher explains the principles and knowledge points accordingly, and the students do the simulation experiment by themselves and are diligent in doing it, so as to achieve the goal of cultivating the practical ability, comprehensive analysis ability and innovation ability.

3.1. Teaching design of port operation in assembly language

When teaching the port programming in Chapter 4 "Assembly Language Programming", the students cannot fully understand the hardware interface technology, but have a certain grasp of the assembly language. With the help of the virtual device "traffic light" embedded in Emu8086, you can first not specifically understand the connection of chips, buses, circuits, etc. related to the hardware interface technology, and focus on the input and output logic operation of the port. At this stage, students only need to understand that "data can be sent to peripherals through OUT command". Emu8086 can well mask the underlying logic, allowing students to focus on the process, syntax, logic, writing, debugging and other operations of assembly language. Of course, at this time, the basic knowledge points in the early stage, such as registers, memory, stacks, and instruction systems, need to be thoroughly understood and mastered.

For this example, the program flow is relatively simple, mainly because the four traffic lights at the intersection are switched in turn and delayed. The simulation of the Emu8086 virtual device is shown in Figure 1. The port address of the traffic light is 4. The 0-12 bits of the word data correspond to 12 lights. When the bit is 1, the light is on, and when the bit is 0, the light is off. The hardware implementation of the specific circuit will be described in the subsequent interface technology.

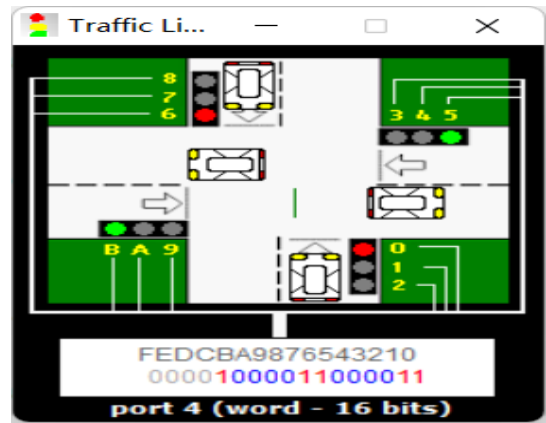


Figure 1. Emu8086 Virtual Device - Traffic Lights

3.2. Emu8086 port operation

The light can be turned on or off through the OUT command, such as:

```
mov ax,01H
out 4,ax
```

Red light No. 0 can be turned on and other lights can be turned off. Teachers can compile demonstration programs, see the running state of programs on virtual devices, and students can intuitively feel and understand the role and running mechanism of instructions. The contents of each register, memory and stack can also be freely observed in the interface, and the operation results are shown in Figure 2.

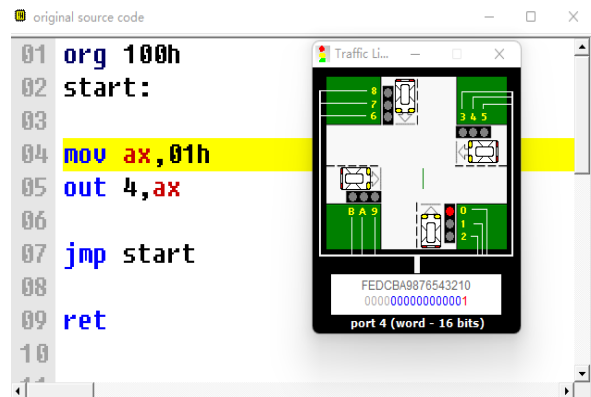


Figure 2. Emu8086 Port Operation

3.3. Design of Emu8086 traffic light control program

The actual traffic light control also requires cycle and delay to control the alternating transformation of lights. To help students master better, teachers need to explain the realization of cycle and delay. The cycle instruction has been mentioned in the previous course, and the delay program has a variety of implementation methods. For example, the delay can be determined by the number of cycles according to the execution clock cycle of each instruction. Assuming that the system uses a 5MHz crystal oscillator, each clock beat needs 0.2us, and then check the clock beat of each instruction to calculate the number of cycles required for the delay. In the 8086 system, the BIOS function call can also be used to realize the delay. For example, in this example, the delay is 5 seconds. The following code can be used:

```
Mov cx, 4Ch
mov dx, 4B40h
mov ah, 86h
int 15h
```

The final program is shown in Figure 3. It can be seen that the editor of Emu8086 is small but powerful. It supports code annotation and is simple and easy to use. What you see is what you get. Students can quickly get started.

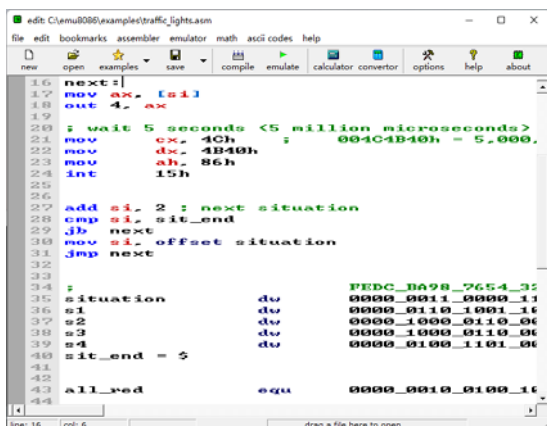


Figure 3. Emu8086 Code Editor

3.4. Emu8086 Debugger

Emu8086 can compile source code to generate executable or binary files. Further, single step tracking debugging can be carried out through EMU8086. The interface of the software is also very friendly and suitable for teaching and learning. As shown in Figure 4, the change status of registers is displayed in real time on the left side of the debugging window, the current machine code is displayed in the middle, the disassembly code is displayed on the right side, and other windows synchronously display screen content, source code, virtual device, stack, flag bit, variable, memory and other information.

The source program can be saved in the standard ASM text format, and compiled to generate COM, EXE or BIN format files. COM files and EXE files can be combined with operation system. COM files are mainly applicable to short assembly language programs because they only have code segments, while EXE files are applicable to complete assembly programs containing data segments, stack segments and code segments. Teachers can introduce and describe different file formats according to teaching tasks and teaching stages.

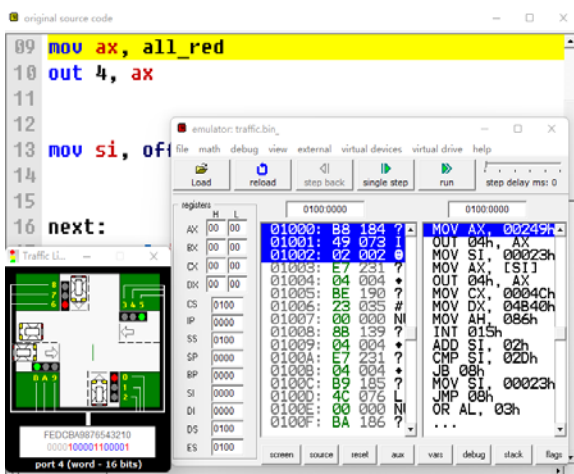


Figure 4. Emu8086 Debugging Environment

3.5. Hardware Practice

Build the hardware circuit with the development board of the experimental box, edit and compile the program in

Emu8086, load the program in the built circuit chip and simulate the operation, and observe the operation results.

The hardware circuit used in the experiment is slightly different from the virtual circuit of Emu8086, and the analog signal lamp needs to be realized with the help of parallel port chip 8255. The Emu8086 integrates the FASM compiler, which can import the compiled program into the hardware firmware. Create a new COM or BIN file in the Emu8086 software and write the following code:

```
org 100h
mov dx,8006h
mov al,90h
out dx,al
mov dx,8002h
mov al,00h
next:
inc al
out dx,al
mov cx,0ffffh
delay: loop delay
jmp next
ret
```

Compile the above code in Emu8086 and generate a BIN file. Burn the BIN file into the firmware to start the experiment.

Through the actual hardware experiment, students not only consolidated the theoretical knowledge of microcomputer principle, developed the programming ability of assembly language, but also mastered the detailed process of hardware development.

4. Conclusion

Microcomputer principle and interface technology is a highly specialized course, which plays an important role in the teaching of engineering majors for application-oriented undergraduate students. With the use of Emu8086 virtual simulation teaching method, excellent teaching results have been achieved. Through the vivid virtual simulation teaching demonstration and students' hands-on experiments, students' enthusiasm for learning was stimulated, the problem of weak practical teaching links was solved, and the limitation of time and space of hardware resources in the laboratory was broken. It provides new teaching methods for teachers, new practical methods for cultivating students' practical ability and innovation ability, and new ideas, methods and means for teaching other courses such as single chip microcomputer principle, embedded system, digital circuit and logic, etc.

Acknowledgment

This work has been supported by Taishan University.

References

- [1] Wang Li."Exploration of Classroom Instruction Method for "Microcomputer Principle and Interface Technology" ".Proceedings of The 2nd International Conference on Politics, Economics and Law(ICPEL 2017).Ed.. Atlantis Press, 2017, 257-260.
- [2] [1]Zhou Ningning,and Deng Yulong."Research and Practice on the Flipped Classroom Teaching Mode in "Microcomputer Principle and Interface Technology" Course Based on the Micro Learning Resources." International Journal of Information and Education Technology 8.3(2018). doi:10.18178 / ijiet. 2018. 8. 3. 1041.

- [3] Rui Liang."Research on Mixed Teaching of "Microcomputer Principle and Interface Technology" Course Based on MOOC".Proceedings of the 2018 8th International Conference on Management, Education and Information (MEICI 2018).Ed.. Atlantis Press, 2018.
- [4] Qing WANG,and Zhong-nian LI."Application for Emu8086 and Proteus in Microcomputer Principle Teaching".Proceedings of 2018 International Conference on Computer,Communication and Network Technology(CCNT 2018 Volume 3).Ed.. DEStech Publications, 2018, 225-228.
- [5] [16] EMU8086 - THE 8086 MICROPROCESSOR EMULATOR 2010. Available: [https:// surendrajat . github.io/ emu8086/](https://surendrajat.github.io/emu8086/).
- [6] Sasko Ristov , “ Using EDUCache Simulator for the Computer Architecture and Organization Course,”INTERNATIONAL JOURNAL OF ENGINEERING PEDAGOGY (IJEP), Vol. 3, no. 3, pp.47– 56, 2013, .
- [7] Milenković Katarina, Stanisavljević Žarko, Đorđević Jovan, “ Visual software system for memory interleaving simulation,” SERBIAN JOURNAL OF ELECTRICAL ENGINEERING, vol. 14, no.1, pp.51-66, 2017.
- [8] Rafael Roman Otero, ,and Alex Aravind."Teaching Computer Architecture Labs using a MCU Platform," Computing Education.Ed , 2018.
- [9] Shilpi Goel, et al, "Verifying x86 instruction implementations," Certified Programs and Proofs, Ed, 2020.
- [10] Ivan Mezei, "Evolution of an educational microprocessor," Computer Applications in Engineering Education, 28.5(2020). doi:10.1002/cae.22300.
- [11] Ying Liu and Fuxiang Gao, "Exploration of Multidimensional Teaching Method Around Assembly Language Programming Course," Proceedings of the International Conference on Education, Economics and Information Management (ICEEIM 2019).Ed. Atlantis Press, 2020.
- [12] ZHANG Jun, TIAN Ze, REN Xiang-long, “ Design and Implementation of Low Power X86 Instruction Decoder,” Aeronautical Computing Technique, vol. 50, no. 5, Sep 2020.