A Web-based CAL System on Computer Architecture and Assembly Language Programming

Yip Kit Kuen Raymond, Li Kei Chun Daniel
Department of Information and Applied Technology, Hong Kong Institute of Education
10 Lo Ping Road, Tai Po, New Territories, Hong Kong
hkryyip@ied.edu.hk danielli@ied.edu.hk

ABSTRACT

This paper introduces a CAL system on Computer Architecture and Assembly Language Programming. The ability of the system is to simulate a computer system that allows the student to understand the whole process from software programming to hardware execution. Multimedia and web-based technologies are applied in the system. The proposed system helps the student to connect theory to practice more easily and can reduce the number of laboratory hours needed.

Keywords: Computer Assisted Learning (CAL), Simulation, Computer Architecture, Assembly Language

1 Introduction

The operating principle of computer system is an important section to all computer, engineering and IT students. It's knowledge usually spread across several modules like Digital Logic Circuit Design, Computer Architecture, Assembly Language Programming and Computer Programming Language. These modules are usually divided into lecture, tutorial and laboratory. Laboratory may be purely hardware implementation (e.g. Digital Logic Circuit Design) or just software implementation (e.g. assembly language programming). Usually, there is no laboratory support for Computer Architecture in small and media size university. This kind of course structure has the following properties

1) The operating principle of computer system is divided into separate pieces (modules). Students are difficult to get an overall picture as there is no laboratory to demonstrates the integration of theories of these modules.
2) The experiments need special hardware (e.g. 68000 microprocessor kit) and software simulator. This leads to difficulties in offering these modules through web-based learning.
3) Student only observes the results rather than the operating process. For example, consider the following 68000 assembly program

Machine Code			Assembly	
Address	Content		Opcode	Operand
00001000	4240		CLR.W	D0

In traditional experiments, student can only observes the content of register D0 is clear and the Program Counter becomes 00001002 when the machine code 4240 (i.e. CLR.W D0) is executed. However, instead just the software result (assembly language programming), the student needs to know the hardware fetch and execute cycle in Computer and CPU architecture. The fetch and execute process of the above assembly is

1. The address 001000(Hex) is output from the CPU through address buses to the memory (RAM).
2. The content 4240 of memory address 001000 is read from memory (RAM) to the Instruction Register (IR).
3. The content of the Program Counter is increase from 00001000 to 00001002.
4. The content 4240 of the instruction register is decoded by the Control Unit.
5. The Control Unit then activates the Clear signal of the D0 register and the content of D0 becomes zero (clear).

2 The CAL system on Computer Architecture and Assembly Language Programming

In 2002, we began to develop a web-based multimedia CAL system on Computer Architecture and Assembly Language Programming. This system simulates the hardware operating process for different software instructions and program structure. The system aims to demonstrates

1) The relationship between programming language, compiler and machine code
2) The function of ALU, Control Unit, Address Register, Data Register, Address Bus, Data Bus, RAM, etc.

Proceedings of the International Conference on Computers in Education (ICCE'02)
0-7695-1509-6/02 $17.00 © 2002 IEEE

3) The fetch and execute cycle
4) Different addressing mode.
5) Stack pointer, subroutine and branch
6) Input, output and interrupt process

The system is divided into 4 areas,

1) Single Assembly and machine code instruction (include different addressing mode), e.g MOVE, ADD, SUB, NOT, AND, OR, CLR, JMP, BEQ (Branch if equal), BNE (Branch if not equal), BSR (Branch subroutine), RTS (Returns), etc.
2) Simple program structure, e.g. If then else conditions, Loop conditions, Subroutine and Returns (including stack pointer), etc.
3) Simple Input/Output Conditions, e.g. Serial IO, parallel IO, Polling, Interrupt, etc.
4) Some simple assembly programs

User can step-by-step run the instruction or program that select fromeach area and observes both the software and the hardware operating process.

In this project, Flash MX is select as the multimedia-editing tool. This is because it provides a rich supports in graphical, audio, video and script programming. Figure 1 shows the first few steps of a PC boot up of the described system.

3 Conclusion

This paper introduces a web-based CAL system on Computer Architecture and Assembly Language Programming. It simulates both the software and hardware process of a computer system. On further expand, the system will allow

1) The user to input and run their own programs.
2) The user to observe the gate level operating process instead of the block level observation (e.g. ALU).

References

[1] Cheng benmao, Zhao jingcheng, "A Multimedia CAL system on Circuit and Electronic Technology Experiments", ICCE, p215-218, 1999.

[2] Sadaf Alam, Roland N Ibbett, Frederic Mallet, Christos Sotiriou, "Computer Architecture Simulation & Visualisation: HASE & SimJava" Institute for Computing Systems Architecture, *Division of Informatics, University of Edinburgh*. http://www.dcs.ed.ac.uk/home/hase/index.html

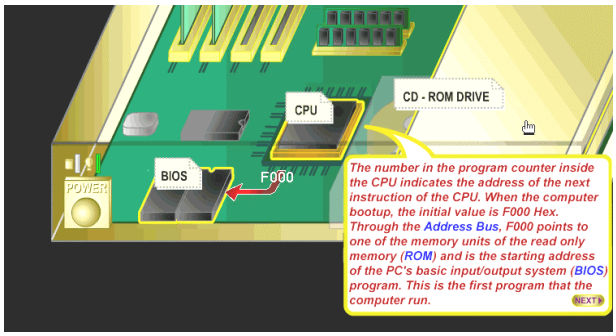

Figure 1a The CPU put the address F000 (Hex) in the address bus that selects one of the memory units of the ROM.

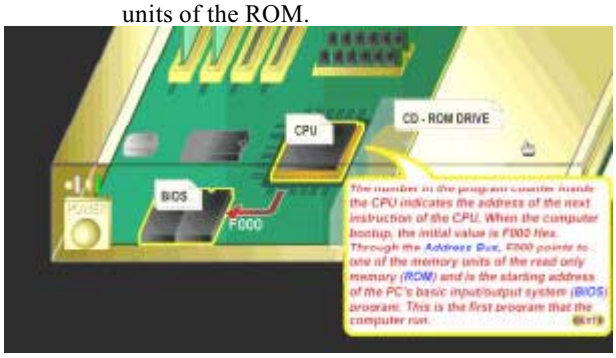

Figure 1b Animation shows the moving direction of F000.

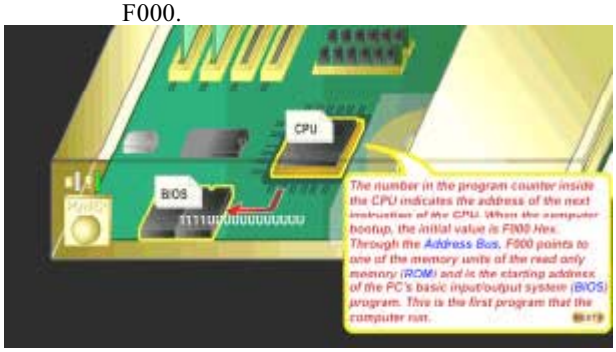

Figure 1c F000 will change to its binary value 1111000000000000 when the mouse is roll over it.

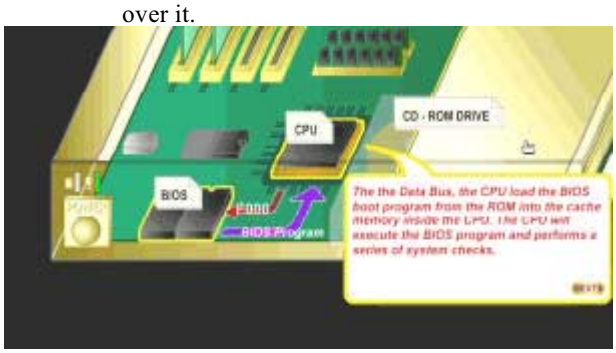

Figure 1d The CPU loads the BIOS program from the ROM.