edu.LMC and Other LMC Simulation Approaches: Contributions to Computer Architecture Education Using the LMC Paradigm

Isabel Pedrosa[1], António José Mendes[2] and Mário Zenha Rela[2]
1 Instituto Politécnico de Coimbra, Instituto Superior de Contabilidade e Administração de Coimbra, Quinta Agrícola, Coimbra, Portugal
ipedrosa@iscac.pt,
WWW home page: http://www.iscac.pt/~ipedrosa/index.htm
2 Universidade de Coimbra, Departamento de Engenharia Informática, Pollo II, Pinhal de Marrocos, Coimbra, Portugal, {toze,mzrela}@dei.uc.pt,
WWW home page: http://www.dei.uc.pt/~toze
WWW home page: http://www.dei.uc.pt/~mzrela

Abstract. The LMC paradigm is not a recent approach to teaching computer architecture: it has been presented, tested and used since 1965, first by its authors, Madnick and Donovan, and their MIT students, and since then in many other universities around the world. The main purpose of the LMC paradigm is to explain, using a very simple model, the main components of a real computer system, and to learn how to program using a simple decimal-encoded instruction set. Using new LMC simulators (based on the LMC paradigm) developed since then, students can nowadays take advantage of simulation processes (e.g., to simulate a program's step-by-step execution). We evaluated six different LMC simulators, picked the "best practices" associated with each one, and developed a new simulator especially focused on management and informatics undergraduate student requirements. This new simulator, edu.LMC, has been tested in a computer architecture course.

1 Introduction

A large collection of simulators for computer architecture students is available nowadays. Web pages [4] and [5] configure repositories of many of those educational resources. However, most of the referenced simulators are not pedagogically adequate, considering the learning context of management and

informatics (M&I) undergraduate students. These students have no more than basic skills in computer architecture. One of the simulators that can be interesting in an M&I learning context is the Little Man Computer, LMC, based on the LMC Paradigm, introduced in 1965 by Stuart Madnick and John Donnovan. LMC simulators are "simple instruction level" simulators and "very simple single accumulator based architecture with a small number of instructions, suitable for a first course in computer science" [6]. The LMC paradigm introduces, through a very simple approach, a model to represent the components of a real computer system, very similar to von Neumann's model. Students can write their own programs using a simple instruction set, and the programs can be tested using an LMC simulator.

In computer architecture courses with M&I undergraduate students at ISCAC[14], we have used the LMC paradigm and an LMC simulator to test LMC programs. LMC simulators allow "students to learn the fundamentals of computer organization/architecture by visually observing and interacting with animated data flow within a particular or real machine" and by "using animated resources" [6].

Several LMC simulators were tested: *Son-of-LMC* and three other directly derived simulators – *FoSoLMC*, *AoFoSoLMC* and *LMC Clone*, Interactive Web-based Simulation and LMC Editor/Assembler/Simulator v1.2.

1.1 The LMC paradigm

As a conceptual approach, the LMC paradigm represents easily understood concepts and features providing a very simple way of understanding the functional areas of a real computer system: ALU (Arithmetic and Logical Unit), Control Unit, Memory, Program Counter and Input/Output Areas. The LMC main area is a mailroom whose fundamental components are: *100 mailboxes* numbered with 2 digits from 00 to 99 (that represent memory in a real system and they are used to store decimal coded numbers corresponding to program instruction, values input by user and operation results), *a calculator* (representing an ALU to store, temporarily, input values or output results and do simple arithmetic additions and subtractions), an *instruction location counter* (program counter, which identifies the instruction being executed at a specific moment), *input* and *output baskets* (I/O, recipients to communicate with outside of the mailroom) and a *Little Man* (control unit, which memorizes instructions and mailbox data and supervises the program execution).

1.2 Methodology

We have tested six LMC simulators concerning their editor, assembler, execution and printing areas, manipulation on input/output and their adequacy in M&I undergraduate computer architecture courses. The decision of choosing those six simulators was based on the fact that they were described in many computer architecture education papers and also due to being effectively used in similar learning contexts. After concluding the LMC simulator analysis, we collected their "best practices" and joined them together in a new LMC simulator, edu.LMC, for special use with M&I undergraduate computer architecture students.

[14] ISCAC – Coimbra Institute of Accounting and Administration

2 LMC simulators

During our research we tested several LMC Simulators: Son-of-LMC (available on http://elearning.algonquincollege.com/coursemat/pincka/dat2343/lectures.f03/14LMC-Simulator.htm) and three other directly derived simulators – FoSoLMC, AoFoSoLMC and LMC Clone; Interactive Web-based Simulation (homepage at http://www.itk.ilstu.edu/faculty/javila/lmc/) and LMC Editor/Assembler/Simulator v1.2 (online at http://www.d.umn.edu/~gshute/cs3011/LMC.html). Table 1 represents a synthesis of that analysis especially considering the most interesting functionalities but also some points that could be improved to create an LMC simulator more focused on pedagogical issues for M&I students.

Table 6. Synthesis of the six LMC simulator analyses.

LMC Simulator	Description	Advantages	Disadvantages
Son of LMC [1]	The basis of other simulators: FoSoLMC, AoFoSoLMC e LMC Clone.	Follows strictly the concepts of the LMC paradigm.	3 digit decimal-encoded instructions; no mnemonics.
Interactive web-based LMC Simulator[1], [2]	This approach has been tested with computer architecture students.	3 areas: Source Program, Opcode, LMC and Program Status Field; mnemonics can be used; it supports 3 different addressing modes.	No example files; No help file.
LMC Editor/ Assembler/ Simulator v1.2	This approach seems to be very effective for M&I students.	Program example files; LMC application window divided into Editor, Assembler and Computer Areas.	Installing process not very friendly for students with no Java skills.

3 edu.LMC Simulator

Collecting all the test results done, we developed a new LMC simulator – edu.LMC, which is available for download at www.iscac.pt/~ipedrosa/LMC/edu_lmc.htm. It consists of a Win32 application, created specifically for students with very basic skills in computer architecture, mostly not majors in computer science or computer engineering. This is a special purpose simulator, with a clear pedagogical focus, tested and used during computer architecture classes for our M&I course. edu.LMC uses instruction mnemonics based on the LMC paradigm, allows instruction commentaries and verifies each instruction before the simulation starts. Students can create, run, debug, print and save programs. This simulator includes example files and a help area describing each instruction. It was our purpose to create a simulator that represents a close approach to the LMC paradigm, including all functionalities and features that are difficult to find together in other simulators. Additionally, edu.LMC is simple to use as our main target users are M&I undergraduate students.

3.1 The application

The application window, shown in Figure 1, is divided into 4 main areas:

- **Editor**: program writing using mnemonics. It's possible to write directly instructions or to open program text files (if they respect the defined structure shown in Figure 2) with a .lmc extension. Programs are verified syntactically as we write each instruction and the operation code (opcode) is generated. The *Load* operation puts *Opcodes* in *Mailboxes*. edu.LMC saves programs, presents the current program name, creates new programs, prints the program and all comments and verifies all syntax errors or inputs that can not be supported.

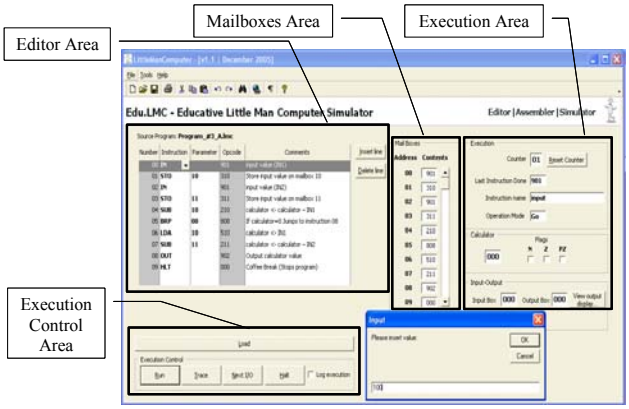

Figure 1. The edu.LMC application window

```
00 IN; input value (IN1)
01 STO 10; Store input value on mailbox 10
```

Figure 2. The edu.LMC input text files.

- **Mailboxes**: mailboxes and its three decimal coded contents. The first are taken for program instructions, the last ones for values stored by programs.
- **Execution**: other functional LMC component contents such as calculator, counter (for instruction location counter), last instruction done (important if there are jumps), instruction name, operation mode, flags, I/O boxes and an option to *view output display* (if output sequence are important).
- **Execution Control**: four modes for executing LMC programs - Run (direct execute from 00 instruction until HALT), Trace (Step by Step), Next I/O (execution stops only for I/O values) and Halt (stops the programs anytime).

Edu.LMC aggregates many functionalities: create, open, comment, save, verify, execute and print programs. Besides that, it includes examples classified by level of difficulty, a complete help area with executable examples of each instruction. Users can also view the saved "Execution Log" file, receive detailed information about program tracing and check or change the Mnemonics Conversion Table.

3.2 edu.LMC core functionalities

- **Execution Log**: Usually students find it difficult to understand the way variable values change. It's possible to activate the "Log Execution" option and generate a text file with information about program execution.
- **Program Tracing Information**: This is another way of tracing program execution. If the execution mode is Trace, a dialog box with information about each instruction is presented, as well as the next one to be executed.
- **Mnemonic Conversion Table**: A significant number of LMC simulators use different instruction codes. This table gives a user the chance to configure those codes or to restore default values.
- **Print Report**: The user can save his programs and/or print them. The Print Report content is very similar to the Editor Window.

4 Conclusions and future work

We have tested six LMC simulators in order to find an adequate simulator for our undergraduate M&I students. Our conclusion was that those simulators included numerous features that are difficult to find together in one simulator. Therefore, we decided to design edu.LMC, a new LMC simulator, simple and user-friendly, and especially focused toward our target students. edu.LMC can be improved by adding functionalities concerning address modes, more examples, a feature to implement array concepts and also an efficient way to represent the fetch-execute cycle.

References

1 Yurcik, W., OsBorne, H., "A Crowd of Little Man Computers: Visual Computer Simulator Teaching Tools", Proceedings of the 2001 Winter Simulation Conference, 2001.

2 Little Man Computer, Illinois State University: School of Information Technology, USA, http://www.itk.ilstu.edu/faculty/javila/lmc/ [online], created on 1998, last modified: 2000-05-01, accessed on 10-07-2005.

3 Yurcik, W., Vila, J., Brumbaugh, L., "An Interactive Web-Based Simulation of a General Computer Architecture", IEEE International Conference on Engineering and Computer Education (ICEDE 2000) San Paulo, Brazil, August, 2000.

4 CAALE - The Computer Architecture and Assembly Language Education Homepage, http://www.sosresearch.org/caale/ [online], last modified: 12-09-2005, accessed on 12-01-2006.

5 WWW Computer Architecture Page – Simulators, http://www.cs.wisc.edu/~arch/www/tools.html [online], last modified: 04-01-2006, accessed on 10-01-2006

6 Cassel, L., Holliday, M., Kumar, D., Impagliazzo, J., Bolding, K., Pearson, M., Davies, J., Wolffe, G., Yurcik, W., Distributed Expertise for Teaching Computer Organization & Architecture, ACM SIGCSE Bulletin, Vol. 33, n.er 2, June 2001, pp. 111-126.