Design of an interactive platform for computer engineering education

HAIDER M. AL-JUBOORI

College of Computer Sciences and Mathematics, Tikrit University, Tikrit, Iraq
E-mail: haider.aljuboori@yahoo.com

ABSTRACT

An interactive software platform, developed at Tikrit University to provide students with a useful training tool for understanding microprocessor arithmetic operations, is presented. The platform design concept is based on the TechCode step-by-step program for problem solving, aimed at testing the understanding of key components presented in the known MUSAID Module. The platform, which was developed under the framework of the eLearning project, is intended to complement rather than replace traditional computer architecture tutorial courses. Its main objective is to design a graphical user interface that provides the end user with maximum control over the presenting process while expecting minimal effort from the presenter programmer. Students feedback has shown promising and encouraging results.

Keywords: E-learning, education technology, software engineering and visualization techniques.

INTRODUCTION

There has been an increasing use of computers as a teaching tool in recent years, especially due to availability of a plethora of interactive computer-based teaching packages that can supplement classroom lectures. However, for some subjects, laboratory work is an integral part of classroom lectures where the subject cannot be assimilated without experimentation. Thus, recent research has focused on integrating simulator and software tool into traditional one-dimensional computer-based teaching tools (Woolf, 2009). For example, Statutor software (Robert, 2011; Pushpa, 2012) is designed to simplify the learning and teaching of statistical concepts, especially those related to sampling distributions based on sampling from a population. Another such initiative is Pegasus (Dekson & Suresh., 2010), which helps students to visualize laboratory work with a 3-phase induction motor. Besides the education technology departments of Universities, some commercial products are also available to facilitate learning of basic principles. DENFORD Machines and Systems Company has developed CNC Desk-Top Tutor for understanding the basic

principles and developing practical skills in Computer Numerical Control 'CNC' (Morozov, 1996). In 2006, Network for Inclusive Distance Education, has developed interactive learning products like Digital Frog International and Snowbird software. They have come up with a novel approach to spread the learning process in an interactive way to disabled students. One such software is "A Digital Field Trip to the Rainforest" which offers self-voicing for users who have visual disabilities. Millard (2000) presents Interactive Learning Modules for Electrical Engineering students which provide interactive and animated simulations, problem sessions and online guidance. Halvorsrud & Hagen (2004) introduced the concepts of Virtual Reality to design a collaborative environment for easy understanding of molecular biology, DNA structure, etc. for secondary students. Woolf (1996) describes technical details of providing interactivity (about use of Flash Animation or JAVA Applets) and conceptual detail of building a learning tool. Shin *et al.* (2002) developed a web-based virtual laboratory system that pioneers an approach of using VRML and XML in the building of simulation and animation. An intelligent multimedia tutoring system has been proposed by Chen *et al.* (2005) for Cardiac Diseases.

The focus of this paper is to present a new adaptive and interactive software platform for technical training and education of computer engineering students on microprocessor arithmetic operations. The main features of this method lies in its interaction and integration within a single package that enables users to access lecture slides, simulate practical demonstrations and conduct assessment using an interactive evaluation system.

DESIGN PROCESSES AND PROGRAM MODELING

Design is "the process of applying various techniques and principles for the purpose of defining a device, a process, or a system in sufficient detail to permit its physical realization". The design process also converts the terminology from the problem space of the requirements to solution space of the implementation (Will & Ken, 1998).

This work describes an exploration model, a corresponding system framework based on that model, and a prototype system that implements the model to achieve its main goal, and to identify and process the special data type for arithmetic operations in an emblematic microprocessor.

A. Model overview

The fundamental components of this model are very generic and can be applied to many other scientific data and visualization systems. However, the model is represented in the context of the Granite Scientific Database System (GSDS) to

simplify the presentation and to clarify the concepts involved with case study.

The abstract of Data Manipulation and Exploration System (DMES) combines an input data set which is implemented using Visual Basic v6.0 (Stallings, 2010). This feature, given the demand for extending the environment with new algorithms, is especially important for the extensive user interaction features (Alexandros & Susanne, 2004; Philip *et al.*, 2002; Aroyo *et al.*, 2006). Figure 1 shows the overall structure of the data manipulation and exploration system (DMES).
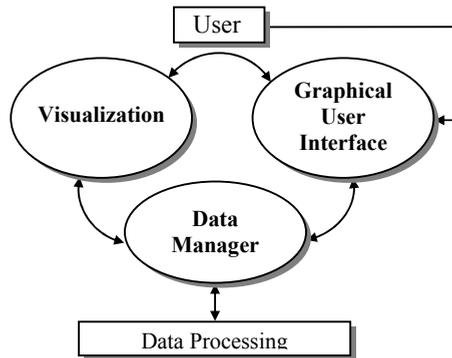
Fig. 1. Conceptual Component Model of Data Manipulation and Exploration System.

The system is composed of three sets of components interfaced with each other. There are seven typical kinds of data processing depending upon the method that the user needs to use.

B. DMES environment

The basic operation of DMES depends on three parts:

- Data manager is responsible for importing and exporting data at the request of the GUI and for translating the simulation data into objects that are usable by the visualization components.

- Visualization component is responsible for retrieving simulation data from the data manager and providing the GUI with visual representations of the simulation data.

- Graphical User Interface component is responsible for presenting the visualization components to the user and for presenting the status of importing/exporting process from the data.

C. Interfacing the DMES components

Through the GUI, the user can specify the simulation data and the status of import/export as shown in Figure 2. Through the Graphical User Interface, the data manager provides the visualization with simulation data and metadata as shown in Figure 3.
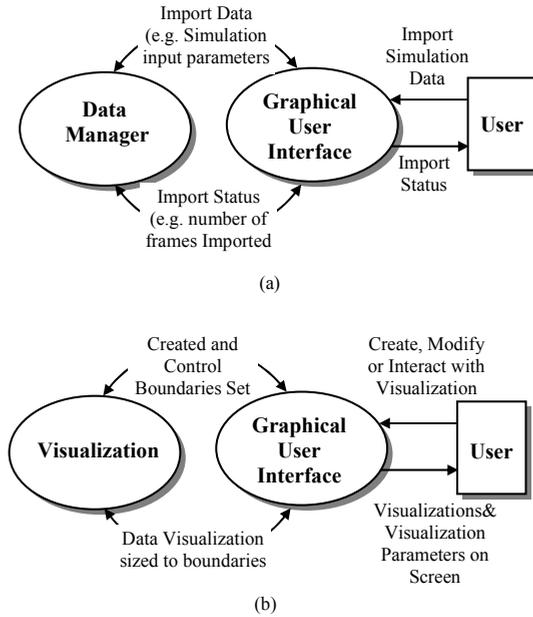

(a)

(b)

Fig. 2. (a) Interface between GUI and importing and managing simulation data components, (b) Interface between the GUI and the visualization.
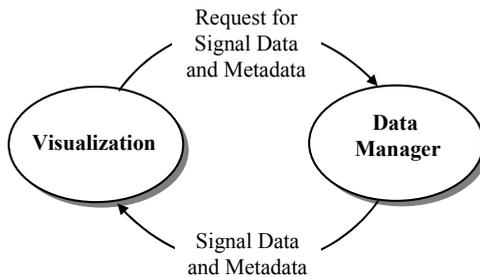

Fig. 3. Interface between the Visualization and Data Components.

IMPLEMENTATION

In order to demonstrate the viability and effectiveness of this model, a prototype system was implemented. The system is a component of a comprehensive integer and floating point arithmetic environment. The model examines the functionality of the Arithmetic and Logic Unit (ALU) in the processor and focuses on the representation of numbers and techniques for implementing all arithmetic operations for binary numbers. Processors typically support two types of arithmetic: fixed point and floating point. For both cases, the model examines the representation of base mathematical operation on numbers and the IEEE 754 floating-point standard.

GUI actions specify both the dataset transformation process and the output parameters in the same interactive window. The philosophy in this framework is to use the GUI to define the transformation techniques that describe the standard arithmetic operations in the CPU. The sequential processes of MUSAID module include seven frames that contain several arithmetic operations concerning binary and decimal numbers (Aroyo *et al.*, 2006; Halvorsrud & Hagen, 2004; Woolf, 1996), such as:

- IEEE 32-bit standard for binary floating-point representation
- Signed binary multiplication (Booth's algorithm).
- Unsigned binary multiplication.
- Convert the decimal number to binary number.
- Convert the binary number to decimal number.
- One's and Two's complements.
- Addition and subtraction of the binary numbers.

Figure 4 shows the major components in interactive window of the main page. In the proposed application, end users have easy and flexible control over the important rendering attributes of each process. The design framework has the explicit display control for all input and output data.
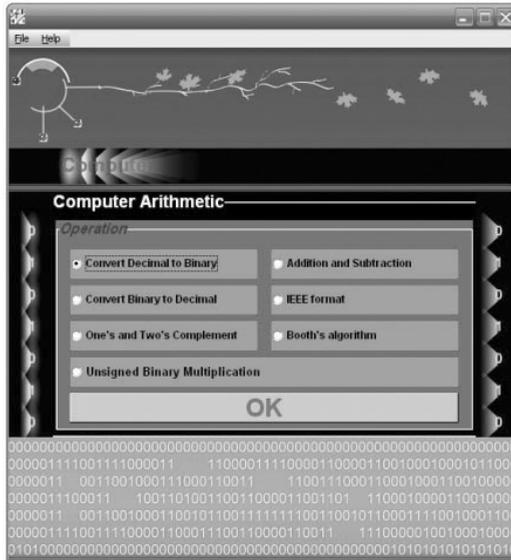
Fig. 4. The interactive window of the main page.

The Booth algorithm will take as case study and the flowchart of twos complement multiplication is depicted in Figure. 5 and it can be described as the same technique as unsigned binary multiplication (Stallings, 2010), the multiplier and multiplicand are placed in the **Q** and **M** registers, respectively.
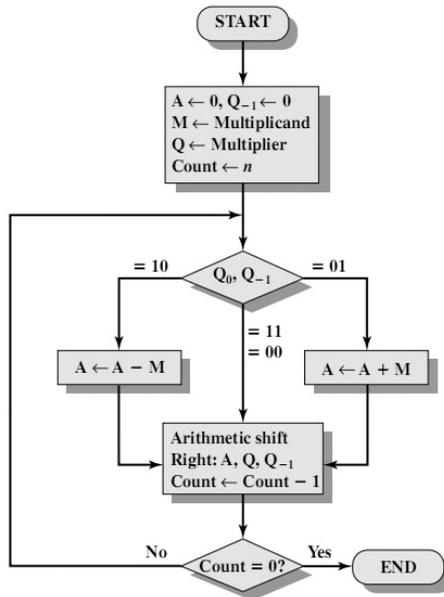

Fig. 5. Booths Algorithm for Twos Complement Multiplication

There is also a 1-bit register placed logically to the right of the least significant bit (Q_0) of the (Q) register and designated (Q_{-1}).

The results of the multiplication will appear in the **A** and **Q** registers. **A** and Q_{-1} are initialized to 0. As shown in Figure 5, control logic scans the bits of the multiplier one at a time.

Now, as each bit is examined, the bit to its right is also examined. If the two bits are the same (**1-1** or **0-0**), then all of the bits of the **A, Q** and **Q-1** registers are shifted to the right 1 bit. If the two bits differ, then the multiplicand is added to or subtracted from the A register, depending on whether the two bits are **0-1** or **1-0.** Following the addition or subtraction, the right shift occurs. In either case, the right shift is such that the leftmost bit of **A**, namely A_{n-1} not only is shifted into A_{n-2} but also remains in A_{n-1}. This is required to preserve the sign of the number in **A** and **Q**. It is known as an **arithmetic shift**, because it preserves the sign bit.

Figure 6 shows the mathematical sequence of events in Booths algorithm for the multiplication of 7 (0111_{binary}) by 3 (0011_{binary}).

```
   A       Q      Q₋₁     M
  0000    0011     0    0111      Initial values

  1001    0011     0    0111      A ← A − M ⎫  First
  1100    1001     1    0111      Shift     ⎬  cycle

                                            ⎫  Second
  1110    0100     1    0111      Shift     ⎬  cycle

  0101    0100     1    0111      A ← A + M ⎫  Third
  0010    1010     0    0111      Shift     ⎬  cycle

                                            ⎫  Fourth
  0001    0101     0    0111      Shift     ⎬  cycle
```

Fig. 6. Booth's algorithm window to determine the sign multiplication of the two binary numbers.

Figure 7 illustrates the same case study using DMES model, where the seventh bottom was selected to present the Booth's algorithm (Patterson & Hennessy, 2005; Stallings, 2010), to evaluate the multiplication process of two sign binary numbers.
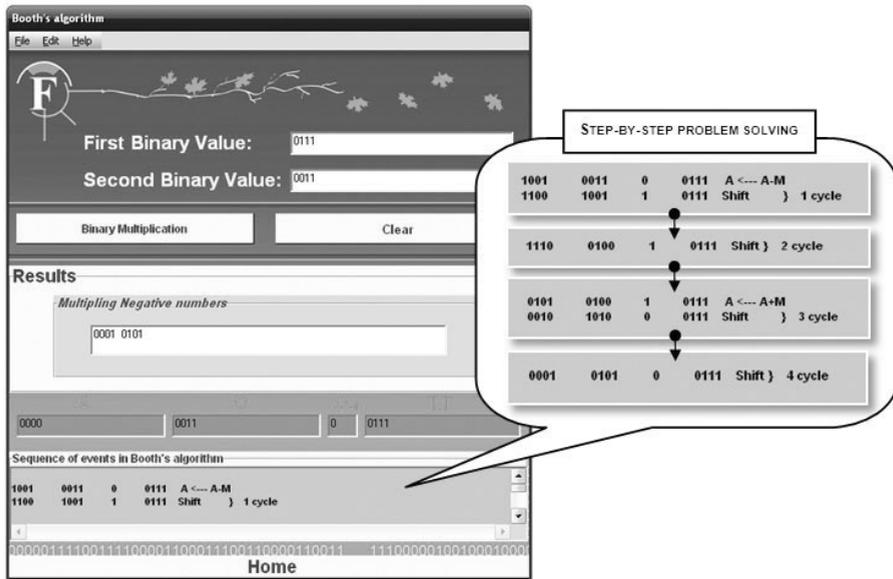
Fig. 7. Booth's algorithm window to determine the sign multiplication of the two binary numbers.Each frame in the results window explains the output process of the current method. The detail cycles for the evaluation of the final results can also be shown in the lowest frame which is referred to as *sequence of events in Booth's algorithm*.

The user facility was added to this TechCode to enable the user to understand the computer architecture concepts using the local help system which was developed by Microsoft Visual Studio 2010 and its associated MSDN Library.
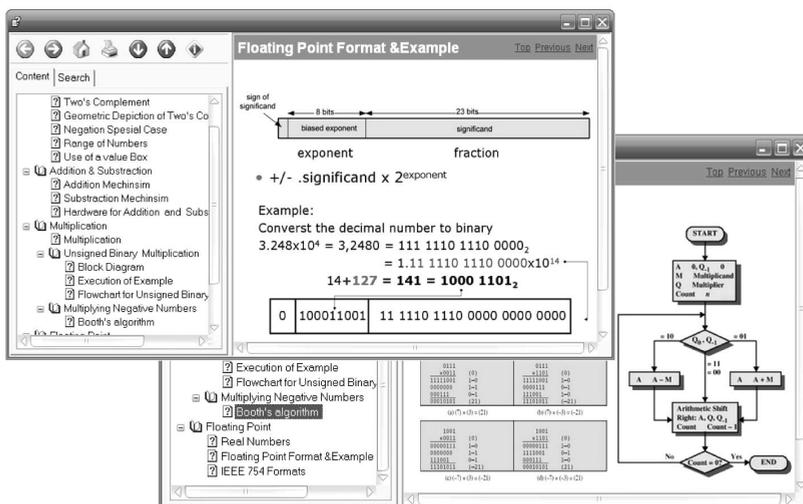
Fig. 8. Demonstrate the interactive windows of online help files.

Figure 8 shows the demonstration window of online help, which is the assist engine, designed by Microsoft Visual Studio.

EXPERIMENTAL RESULTS

The main purpose of this technique is to provide a well-defined systematic interactive problem solving that will help in testing the users understanding of key concepts in each main section of the computer engineering curricula. A survey for the Beta version of the MUSAID module was conducted on students with different majors (the principle in computer architecture and organization) at Tikrit University. The results were analyzed to measure students feedback. The overall assessment of the class, who tried the eTrainer, resulted in an average score of 77.8% for students who believed that the eTrainer technique was beneficial and 3.7% for non-effectiveness of the tool. The rest of the students were neutral. In addition, 79.3% of the students believed that the Code was a good tool for improving the understanding of the major concepts in the class, while 17.3% thought that this tool was not the only reason behind the understanding of the concepts but the module was good and user friendly. Only 3.4% of the students reported that the tool was not effective for them. Figure 9 shows the survey results. Most answers were positive, and the students expressed their willingness to use this tool for all their courses. Most students welcomed the new tool, which reflected the importance of these on-line modules.
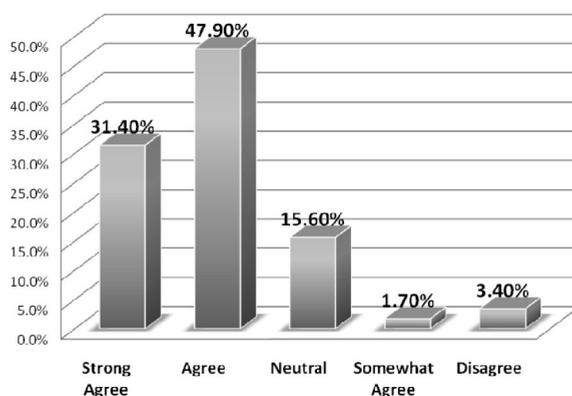

Fig. 9. Survey results of students opinions.

CONCLUSION

A new interactive and adaptive method to maximize students learning experience in their self-learning process and increase their understanding of

microprocessor operations, was presented. The aim was to provide a blueprint for building a visualization and data exploration framework to serve computer operations technique, architecture, and organization education. The tool was based on MUSAID module that guides the user through a step-by-step analysis of arithmetic operation in conventional computer system CPUs. The algorithm and design process were flexible and can be further developed to incorporate new ideas and methods in the fields of computer architecture and organization, in addition to its use in programmable logic. The model can also be employed in laboratory education of undergraduate students.

The proposed method may pose as a best-practice experiment leap in engineering education and may facilitate learning at a pace specific to the learner without the constraints of a fixed time-span i.e. semester or quarter, currently specified in computer engineering curricula. The work will be continued to improve the ability of recognizing the different learning approaches for different content and users. In addition, the model can be modified to include the pipeline operation and superscalar technique inside the microprocessors.

ACKNOWLEDGEMENT

REFERENCES

Alexandros, P. & Susanne, L. 2004. Adaptive Learning Environments and e-Learning Standards. Electronic Journal on e-Learning 2(1): 181-194.

Aroyo, L., Dolog, P., Houben, G.J., Kravcik, Naeve A., Nilsson, M. & Wild, F.2006. Interoperability in Personalized Adaptive Learning. Educational Technology and Society 9(2): 4-18.

Chen, C., Lee, H. & Chen, Y. 2005. Personalized e-learning system using Item Response Theory. Journal of Computers & Education 44(3): 237-255.

Dekson, D.E. & Suresh, E.S.M. 2010. Adaptive E-learning Techniques in the Development of Teaching Electronic Portfolio - A Survey. International Journal of Engineering Science and Technology 2(9): 4175-4181.

Halvorsrud, R. & Hagen, S. 2004. Designing a Collaborative Virtual Environment for Introducing Pupils to Complex Subject Matter. *Proceedings of the third Nordic conference on Human-computer interaction.*

Millard, D. 2000. Interactive Learning Module for Electrical Engineering Education, *Proceedings of the Electronic Components and Technology Conference, pp. 1042- 1047.*

Morozov, E. 1996. Implementation of Computer Based Teaching Systems for professional Training in Computer Aided Engineering. *Proceedings of the ICDED96.*

Network for Inclusive Distance Education 2006. Interactive Learning Tools; Available at:http://nide.snow.utoronto.ca/Inter-activeindex.html.

Patterson, D. & Hennessy, J. 2005. Computer Organization and Design: THE HARDWARE/SOFTWARE INTERFACE, Elsevier, San Francisco.

Philip, J., R. Daniel, B., & Ted, M. 2002. A Data Model for Distributed Multisource, Scientific Data. Hierarchical and Geometrical Methods in Scientific Visualization, Springer-Verlag, Heidelberg.

Pushpa, M. 2012. ACO in e-Learning: Towards an adaptive learning path. International Journal on Computer Science and Engineering (IJCSE) 4(3): 458-462.

Robert, W. 2011. Statutor Version1.23 A computer-based teaching tool for statistical concepts. Available: http://archives.math.utk.edu/software/msdos/statistics/statutor/statu123.readme.

Shin, D., Yoon, E., Lee, K., & Lee, E. 2002. A Web-based Interactive Virtual Laboratory System for Unit Operations and Process Systems Engineering Education: Issues, Design and Implementation. Computers and Chemical Engineering, 26(2): 319-330.

Stallings, W. 2010. Computer Organization and Architecture; Design for Performance, Pearson Education Inc, New Jersey.

Will, S. & Ken, M. 1998. The Visualization Toolkit, 2nd Edition, Prentice-Hall.

Woolf, B. 1996. Intelligent Multimedia Tutoring System, Communications of the ACM 39 (4): 30-31.

Woolf, B. 2009. Building Intelligent Interactive Tutors Student-Centered Strategies for Revolutionizing E-learning, Morgan Kaufmann Publishers/ Elsevier.

تصميم برنامج تفاعلي لتعليم هندسة الحاسب الآلي

حيدر مهدي الجبوري

كلية علوم الحاسوب والرياضيات – جامعة تكريت – تكريت – العراق

خلاصة

يسهم هذا العمل في تقديم وصف لبرنامج تفاعلي تم تطويره في جامعة تكريت بغية تزويد الطلاب بأداة تدريبية مفيدة لفهم العمليات الحسابية الخاصة بالمعالجات الدقيقة. ويستند تصميم هذا البرنامج على مفاهيم البرمجة والتعليم خطوة بخطوة (TechCode) لحل المشكلات، حيث يهدف الى اختبار مستوى فهم العناصر الأساسية بما يعرف بنموذج "مساعد". يهدف النظام الأساسي، الذي تم تطويره في إطار مشروع eLearning، ليكون تمهيدا لاستبدال بعض الاجزاء التقليدية للمحاور التعليمية في مادة المعمارية وهندسة الحاسوب.ترتكز الغاية الرئيسية لهذا المشروع على تصميم واجهة استخدام رسومية توفر للمستخدم أقصى قدر من السيطرة على عملية التقديم، بينما نتوقع اقل قدر يمكن من الجهد يبذله المبرمج. الى جانب ذلك نوقشت في هذا المقام ردود أفعال الطلاب وآرائهم كجانب مهم في عَملية تحسين وتطوير البرامج التعليمية.