

The Classroom Computer: A Role-Playing Educational Activity*

*Kenneth H. McNichols, M. Sami Fadali
Electrical Engineering / MS260
University of Nevada, Reno
Reno, NV 89557*

Abstract

Computers are among the most exciting technical systems for younger students. Many students are keenly interested in learning how a modern computer works, but are unable to because of their limited background. We have developed a series of activities that mimic the operation of a digital computer to teach computer architecture concepts without the need for expensive equipment that are beyond the resources of most schools. Students role-play parts of a digital computer to accomplish a given task, and follow a given set of rules (their program). Student roles include: a processor, a cache memory controller, main memory, mass storage devices, system busses, and input/output devices. Student activities include displaying a multimedia movie, exploring cache memory, and processing an image. Preliminary testing indicates that the Classroom Computer allows students to understand the basic operations of a digital computer.

I. Introduction

The US needs to graduate a sufficient number of engineers to effectively compete with other industrialized nations [1]. The importance of technology to the nation is reflected in the prominence that it is given in the recent *National Science Education Standards* [3]. Unfortunately, at this stage K-12 education in most states does not sufficiently prepare students for careers in engineering and technology and does not promote them as viable career options. With partial support from NSF, the College of Engineering at the University of Nevada has established a Mobile Engineering Education Lab (ME²L) to promote engineering and technology in northern Nevada schools. The lab involves K-12 students in activities that explain the operation of a variety of engineering systems that appeal to the students and teach some aspects of engineering practice. In addition, an engineering capstone course [2] was designed for future high school science and mathematics teachers to promote engineering in their classes. This paper presents a student activity that was developed to explain computer structure and operation for use with ME²L and with the capstone class for high school teachers.

* Partial support for this work was provided by the National Science Foundation's Division of Undergraduate Education through grant DUE #1970638.

Computers are among the engineering systems that are known and liked by most K-12 students. They use them in classroom learning activities, to prepare assignments and reports, and to play a variety of games. Students are often curious about how computers accomplish the many tasks they use them for. However, few students have the background needed for a rigorous study of computer architecture. We have developed a series of role-playing activities that mimic the operation of a computer based on a simplified computer model. The activities allow the students to understand the operation of the computer by playing the roles of computer subsystems. We have dubbed these learning activities the Classroom Computer.

In each activity a group of students play the roles of computer subsystems, and collectively accomplish a given task similar to those of their own computers. The tasks students accomplish are high level, and are intended to demonstrate the ideas of computer architecture and data flow with an increasing level of complexity for higher grade students. Tasks are completed by having the students move blocks of data through the computer, but do not necessarily require students to perform any of the calculations. The activities allow students to study important computer concepts without requiring them to go through complex instructions to understand the programming of their tasks. A University student volunteer is in charge of directing the learning activities and providing instructions for the students.

A similar project "The Computer in Action" [4] has been developed to show students how a computer works. "The Computer in Action" is intended for primary education students, and emphasizes the reliance of computer operation on given set of instructions (program). Our activities avoid the use of programming and processor instructions whenever possible to allow students to focus on high level tasks and learning objectives.

This paper describes the computer role-playing activities as well as student reactions and responses to them. We first discuss the model used to develop all activities, and student roles. Three sample activities used in The Classroom Computer are presented with descriptions and instructions. Results from testing on beginning computer engineering students at the University are also presented. Finally, some future activities and other applications of the role-playing concept are given.

II. Computer Model

The Classroom Computer activities teach students the basic architecture principles underlying the operation of a modern digital computer using components commonly found in their own personal computers. We base our activities on two simple computer models adapted from introductory computer engineering texts [5], [6]. Less complicated activities use a single-tier bus architecture model shown in Figure 1. More complicated activities for higher grade students use the two-tier bus architecture shown in Figure 2. For specific activities some components, such as Cache Memory, may be omitted from the model and others, such as input/output devices, may be added to it.

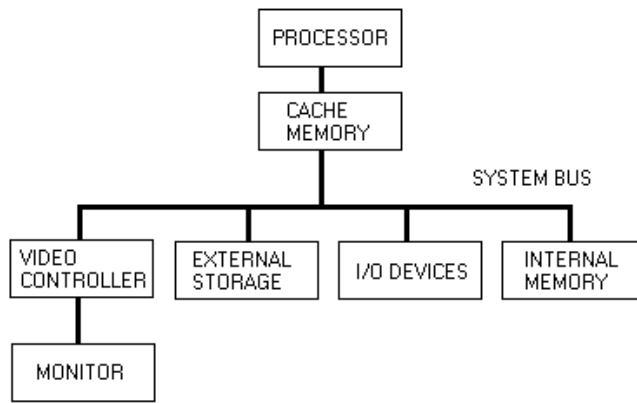


Figure 1: Single-Tier Computer Model

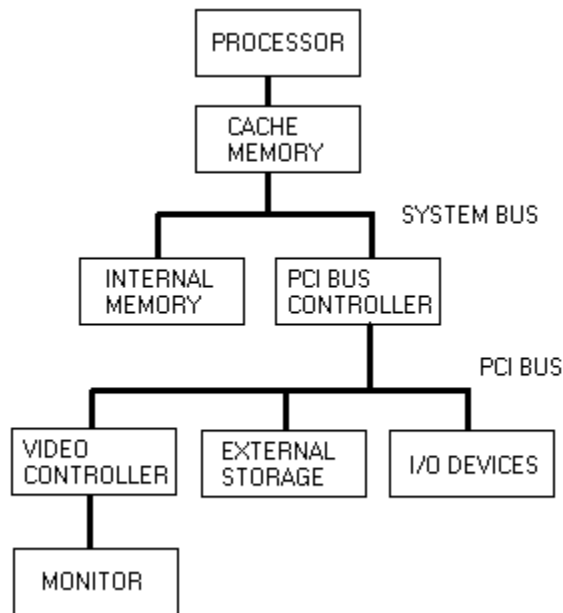


Figure 2: Two-Tier Computer Model

Students perform the roles of the following computer subsystems:

- 1) *Processor*: The Processor performs all calculations for the computer, and directs the operation of the other components.
- 2) *System Bus*: A bus Transfers data from one subsystem to another within the computer upon command from the Processor. After each operation is complete the System Bus returns to the Processor for a new command.
- 3) *Cache Memory*: Cache Memory holds blocks of data very close to the processor so those individual data elements can be accessed more quickly.
- 4) *Internal Memory*: Internal Memory holds a limited amount of data for the computer.

- 5) *External Storage*: External storage devices store large amounts of data for the computer in bulk. External storage is very slow in comparison to the speed of internal memory.
- 6) *PCI Bus*: The Peripheral Component Interconnect (PCI) Bus transfers data between the computer's system bus and any external devices used by the computer.
- 7) *PCI Bus Controller*: The PCI Bus Controller directs the operation of all the external devices attached to the PCI Bus.
- 8) *Monitor Pixels* : Pixel elements, or Pixels, hold a very small piece of an image as part of a display device like a monitor.
- 9) *Video Controller*: The Video Controller is in charge of the contents of the Monitor. The Video Controller gives each Pixel a data element to display.
- 10) *I/O Devices*: Input/Output (I/O) Devices such as a mouse, keyboard, or image scanner are used to transfer information into and out of the computer.

In the sequel, reference to a capitalized computer subsystem such as "Processor" actually refers to the student playing the role of the referenced subsystem. Non-Capitalized reference to a subsystem such as "processor" refers to the computer subsystem itself.

III. Student Activities

In this section, we discuss sample student activities that explain specific tasks performed by computers. For each activity, we give a brief general description, a list of student roles, an overview of the role-playing procedure, and an itemized list of the steps in the procedure.

III.1 Display A Movie: Basic Architecture Concepts

Grades 7-12

Description - Students display a multimedia movie of 3x3 image frames. This activity shows students how basic computer components communicate, and how the computer can represent the data for a movie. The single tier computer model of Figure 1, without Cache Memory or I/O Devices, is used to display the movie. Upon completion of the activity students are able to describe the roles of a computer processor, bus, internal/external memory, and show how a computer represents a movie as a series of images.

Student Roles - Students play the parts of Processor, System Bus, External Storage, Internal Memory, Video Controller, and Monitor Pixels (0-9).

Procedure Overview - The data for each image frame is kept by External Storage. Students move the data in image frame blocks from External Storage to Internal Memory. Each frame block includes nine Monitor Pixel data elements (pieces of

colored paper.) The frame blocks are composed of nine pieces of. Students then move the data elements for each Monitor Pixel from Internal Memory through Processor to Video Controller, and finally to Monitor Pixels. Processor must assign a pixel number to each data element after it is taken from Internal Memory.

Procedure

1) Acquire Data From Storage

Processor sends System Bus to retrieve the movie data from External Storage. System Bus accepts a box containing the data for the first frame from External Storage, and gives it to Internal Memory who labels and stores the frame. This process repeats until there are no more frames in External Storage.

2) Acquire a Frame from Memory

Processor directs System Bus to retrieve the next (first) frame from Internal Memory. System Bus then goes to Internal Memory and asks for the next image frame. Internal Memory gives the frame to System Bus who delivers it to Processor. If there are no more frames Internal Memory informs System Bus who informs Processor.

3) Assign Pixel Numbers to Data

Processor takes each piece of data from the box in order and writes a Pixel number on the back of the card.

4) Display Frame

Processor gives the block of data to System Bus and tells System Bus to give the image data to Video Controller. System Bus delivers the data, and returns to Processor for the next instruction. Video Controller takes each piece of data out and gives it to the corresponding Monitor Pixel who then holds his/her element for display.

5) Repeat

Return to Step 2, and repeat until there are no more frames.

III.1 Exploring Cache Memory

Grades 8-12

Description - Students find the average of a series of 15 numbers. This activity shows how cache memory is used to make computer calculations faster, how a bus is used for transferring data, and how internal/external memories are used for storing data. The computer model shown in Figure 1, and a memory hierarchy shown in Figure 3 are used. Time permitting the activity may be repeated without Cache Memory to show that calculation time is decreased when using cache. Upon completion of the activity students can describe the roles of a processor, internal/external memory, and the purpose of cache memory.

Student Roles - Students play the parts of Processor, Cache Memory, Monitor, Internal Memory, and External Storage. This activity does not require the use of Pixels for the display, so the Video Controller is omitted from the computer model.

Procedure Overview - The data used in the activity is kept by External Storage in three blocks of five numbers. Students move the data from External Storage to Internal Memory. Students then move the blocks of data from Internal Memory to Cache Memory where Processor can access the data elements one at a time to make calculations. Cache Memory can keep a maximum of two blocks of data at any time. If the activity is repeated without using Cache Memory then Processor must instruct System Bus to get the elements from Internal Memory one at a time.

Procedure

1) Move Data From External Storage to Memory

Processor asks System Bus to move all the blocks of data from External Storage to Internal Memory. System Bus gets the blocks from External Storage one at a time and gives them to Internal Memory. Once this is complete, System Bus returns to Processor for the next instruction.

2) Fill Cache Memory from Internal Memory

Processor tells System Bus to take the first two blocks of data from Internal Memory and give them to Cache Memory. System Bus performs this action and returns to Processor.

3) Calculate and Keep Cache Filled

Processor takes the data from Cache Memory and enters them in some adding device (usually a calculator). When one of Cache Memories' blocks is empty Processor directs System Bus to get the next block of data from Internal Memory and give it to Cache Memory.

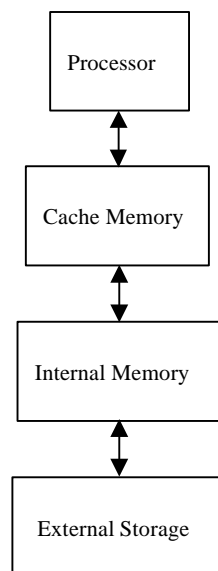


Figure 4: Memory Hierarchy Model

4) Repeat

Continue with step three until Internal Memory informs System Bus that there is no more data.

5) Find the Average

Processor divides the final number in the adder by the total number of data items. Processor keeps track of the number of data items by counting input operations or the number is initially given to Processor.

6) Output Results

Processor writes the average on a card and instructs System Bus to take the card to Monitor.

III.3 Image Processing: Computer Architecture

Grades 10- 12

Descriptions - Students convert an existing image to an array of pixel elements, display, manipulate, and redisplay the image. This activity shows how a computer stores an image and how data moves through a computer's layers of busses. The computer model shown in Figure 2 is used for this activity. Upon completion of the activity students will be able to describe the tiered architecture of computer busses and how an image is stored, and manipulated by a computer.

Student Roles - Students play the parts of Processor, System Bus, Internal Memory, PCI Bus Controller, PCI Bus, Video Controller, Monitor Pixels (0-11), and Image Scanner. Image Scanner is an I/O Device specific to this activity. External Memory and Cache Memory are omitted from this activity.

Procedure Overview - Students are shown a 16-color grayscale image on a large piece of posterboard. Folders containing two copies of digitized data for each pixel element are kept by Image Scanner. Each copy is a two-sided piece of paper with original data on the front and modified data on the back. Students move the data from each element from Image Scanner through PCI Bus & System Bus to Internal Memory. They then send the first copy of each data element to Monitor Pixels. Processor is in charge of controlling the movement of data and labeling each data element for the correct Monitor Pixel. After this is complete students modify and display the same data elements.

Procedure

1) Scan The Image

A Student shows the image posterboard to the class. Image Scanner takes the image and picks up a box containing the digitized data.

2) Store Image in Memory

Image Scanner signals PCI Bus Controller verbally that there is new data and PCI Bus Controller sends PCI Bus to Image Scanner to accept the first data Element. PCI Bus takes this data to PCI Bus Controller. PCI Bus Controller signals Processor verbally that there is new data and Processor sends System Bus to retrieve the data. PCI Bus Controller gives the first element to System Bus and sends PCI Bus back to Image Scanner for the next element. Simultaneously, System bus takes the element to Processor. Processor instructs System Bus to give the Data Element to Internal Memory for storage. System Bus delivers the element to Internal Memory who stores the element in order. System Bus then returns to Processor who tells System Bus to get the next element from PCI Bus Controller.

3) Repeat Until All Elements Are Stored In Memory

When Image Scanner is out of data element they inform PCI Bus that there is no more data. PCI Bus returns to PCI Bus Controller and tells PCI Bus Controller that there is no more data. When System Bus comes for a data element PCI Bus Controller tells System Bus there are no more elements. System Bus returns to Processor.

4) Display Image

After all the element blocks are stored Processor directs System Bus to retrieve the first/next pixel element from Internal Memory. System Bus goes to Internal Memory who takes the first copy from each element block and gives it to System Bus. System Bus takes it to Processor who labels it for the appropriate Monitor Pixel then gives it back to System Bus with instructions to take it to Display Controller. System Bus takes the element to PCI Bus Controller and returns to Processor. PCI Bus Controller gives the element to PCI Bus to take it to Video Controller. Video Controller gives the element data to the corresponding Monitor Pixel while PCI Bus returns to PCI Bus Controller. This process continues until there are no image elements to display at which time System Bus Returns to Processor for further instructions.

5) Modify the Image

Processor instructs System Bus to retrieve the image elements one by one from Internal Memory. System Bus gets the first/next image element from Internal Memory and returns it to Processor. Processor inverts the element by flipping it over and gives it to System Bus to return to Internal Memory. The process continues until there are no more image elements to modify.

6) Display the Modified Image

Repeat step 4) with the modified image data.

IV. Test Results

Although the Classroom Computer is intended for a younger audience, at this stage we have tested the concept using beginning computer engineering university students. We are planning to test the Classroom Computer in visits of ME²L to Northern Nevada schools and will include our findings in the final version of this paper. Our tests indicate that while The Classroom Computer accomplished the goal of teaching computer architecture and data flow concepts several issues need to be addressed. We originally intended to make the activities work with a minimum of instructions to students. This approach confused the students participating in our tests and made the activities unappealing. The solution is to give the student volunteer a more active role in the activities and to transform the activities into a kind of play for the students. Each student will be given a script describing the role of their "character" and a set of lines. The University student volunteer will become a stage director providing cues and leading students, and/or data, from place to place during an activity.

A final problem we encountered during testing was that students became bored when either only one student worked, or when the activity became too repetitive. For example, in Display a Movie (Section III.1) we originally used five 4x3 image frames and had students transfer data for an entire image frame from Internal Memory to Processor to Video Controller. This became a problem after two or three frames because Processor dominated the activity and students understood its goals after one or two iterations. The solution was to use fewer small image frames and transfer data one element at a time. Similar problems were encountered in Exploring Cache Memory (III.2) and again the solution is to reduce the size and number of data blocks.

V. Conclusions & Future Work

Role-playing can be invaluable for teaching students concepts when experimentation is not an alternative, when experiments for a given subject become prohibitively expensive, or when students do not have the educational background to understand an experiment. The Classroom Computer gives students a rudimentary understanding of computer architecture, and how data moves through their computers. In the near future we will transform our existing activities into plays for the students to act out in a small time period. We will then complete more tests using University students, and begin tests in local secondary schools. Some other possible applications of role-playing include teaching programming data structures, data transfer in computer networks, and introducing students to automatic control systems.

VI. References

- [1] Advisory Committee to the NSF Directorate for Education and Human Resources, *Shaping the Future: New Expectations for Undergraduate Education in Science, Mathematics, and Technology*, NSF 96-139, NSF, Arlington, VA, 1996.
- [2] Robinson, M., Fadali, M. S., "A model to promote the study of engineering through a capstone course for pre service secondary science and mathematics teachers", *Proc. FIE' 98*, Tempe, AZ, Nov. 1998.
- [3] National Research Council. *National Science Education Standards*. Washington, D. C.: National Academy Press, 1996.
- [4] Addison, P., "The Computer in Action", <http://www.cs.purdue.edu/outreach/workshops/Action.html>.
- [5] Stallings, W., *Computer Organization and Architecture*, Prentice Hall, 1993.
- [6] Tanenbaum, A., *Structured Computer Organization*, Prentice Hall, 1999.