How Computer Animations Make Teaching Complex Topics More Effective And More Efficient

Stephen J. Kuyath

UNC-Charlotte Department of Engineering Technology

Abstract:

Complex or abstract topics are sometimes very difficult for students to understand. Many of these topics can be taught in a more efficient and effective manner. Computer animations, both user-controlled and free running, with audio or textual clarification, appeal to a number of learning styles that will help students master the subject. The methods used grew from teaching an introductory course in microprocessors. Students had a difficult time understanding the abstract concept of addressing modes. In a typical semester, up to three weeks of instruction and many example programs would be necessary to cover this topic. Several user-controlled animations with textual explanations were developed to create a course module on addressing modes. The instructor controlled the animations as he explained the concepts in the traditional classroom. An audio explanation was then included in the animations and the package was placed on the web to provide student access for review. Using the animations in class, the instructor was able to provide immediate feedback to the students as questions were raised. This also provided the feedback necessary to improve the animations before they were placed on the web. The advantages to creating the animations were: 1) More efficient use of class time. Less class time was used to introduce addressing modes so that more time was available to discuss advanced topics; and 2) Better student comprehension. Students' grades and projects improved noticeably. The final product took time to develop. Methods used to develop the animations and some lessons learned will be discussed. Data will be presented, based on students' grades and opinions that will show that the addition of web-based computer animations made the course more efficient and effective.

Introduction:

For many years the author taught introductory microprocessor courses and found that many students had a difficult time adjusting from the more concrete subjects such as introductory circuit analysis to topics as abstract as microprocessors. Complex microprocessor concepts, such as read and write cycles, addressing modes, and instruction fetch and execute cycles, were very difficult for students to comprehend. A traditional lecture, with chalk and blackboard, was clearly not as effective as it could be. Many methods were attempted to teach this material, including simulations, hands-on activities, and demonstrations, but the method that worked best was the application of user-controlled animations.

In his paper, “Reaching the Second Tier: Learning and Teaching Styles in College Science Education”, Richard Felder [1], suggests that learning styles may be broken down into five basic categories: 1) Sensory or Intuitive, 2) Visual or Verbal, 3) Inductive or Deductive, 4) Active or Reflective, and 5) Sequential or Global. Felder also indicates that many faculty use a teaching style that, although very comfortable to the faculty, leans heavily toward a very small percentage of the students. The consequence is a decline in enrollment in science education at a time when our nation needs more science, engineering, and technology graduates. Felder argues that if faculty would employ teaching methods that address all the basic learning styles at least some of the time, that students would have more interest in the topics being discussed, and thereby gain better understanding.

The Problem:

As in many science and engineering courses, there are topics that must be introduced to the students before they can begin to understand the bigger picture. In an introductory microprocessor course one of these difficult topics is addressing modes. The author believed that the topic of addressing modes could be covered more efficiently, saving time for more advanced topics later in the semester. The problem was that students were having a very difficult time understanding this topic, and it was preventing them from fully comprehending later topics in the course.

A further complication was that the ability to program in assembly language was one of the main objectives of the course. The author felt that students must fully understand addressing modes and assembly language programming before proceeding to more advanced topics. Many assignments and projects built upon the students’ ability to program in assembly language. Mastery of assembly language programming was required for students to be successful in this course and in the advanced courses in microprocessor systems design and computer engineering technology.

Solutions:

After researching some interesting methods and experimenting with the Internet it was decided that the best way to explain addressing modes was through the use of computer animations. It was believed that if complex topics were explained through animations, multimedia, and interaction, student understanding would be greatly improved [2].   Some simple line drawings were created to step students through the processes that the microprocessor uses to address memory. A JavaScript program was used to combine the images into an animation as it allows the students to step through the animation at their own pace. Figure 1 shows a sample image of the first frame of an animation. As shown in Figure 1, this is not a sophisticated image. Simple line drawings were chosen for four reasons:

1. Simple images were all that was needed to relay the information,
2. The tools needed to create these images were readily available,
3. Simple Images download faster,
4. Simple images are easier and quicker to create.

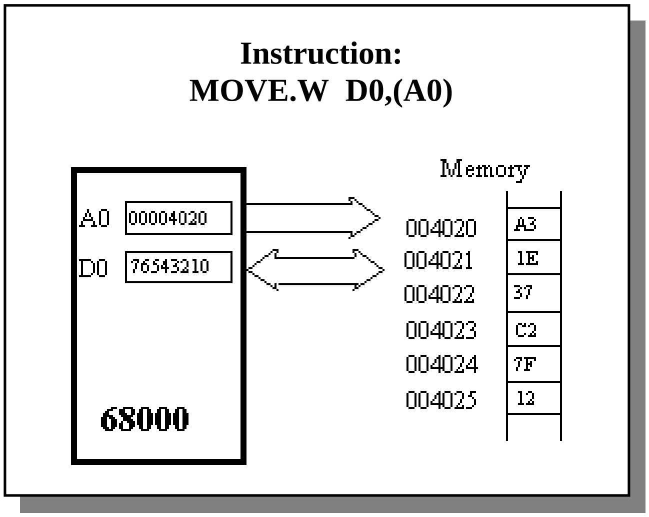


Figure 1: First frame of the Address Register Indirect Animation

The first image simply sets the stage. Subsequent images show the steps as the microprocessor places the address on the address bus, data on the data bus, and writes the data to the specified address (see Figure 2).

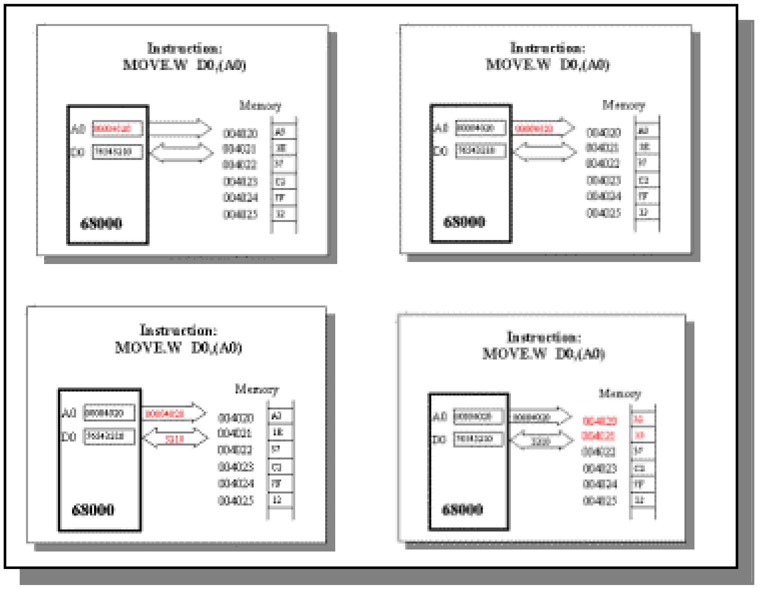


Figure 2: Subsequent Images of the Address Register Indirect Animation

Initially, this animation was displayed in class as the instructor explained each step of the process. Several benefits were immediately noticed:

1. Less set-up time was needed to draw the images on the board,
2. The author could easily step forward or backwards through the animation,
3. The author covered approximately three times the material in one lecture,
4. Students seemed to understand the information much more quickly than in previous semesters.

Students also seemed to be paying closer attention in class. After students were told that the animations were on the web site and that they could access them whenever they chose, they put down their pencils and focused on the topic under discussion

Implementation Results:

The students understood the information much quicker that in previous classes. The topic of addressing modes, which previously had consumed as many as three weeks of lecture, was reduced to a week. This allowed two weeks at the end of the semester for a more involved project or for the discussion of more advanced topics.

The students' understanding and grades also improved. It is assumed that this is due to the many learning styles addressed during the lecture. The animations were sensory, visual and verbal, inductive, active and sequential. According to Felder [1], this is one of the best possible combinations for student learning.

Figure 3 shows the averages class grade for the four tests routinely assigned in this course. The first test shows that all three groups were typical, with averages between 68 and 72. In a typical lecture class, without the use of the animations, the test average dropped significantly on the second test – the test covering addressing modes. The classes using the animations scored significantly better on the addressing modes test and then performed better throughout the remainder of the class.

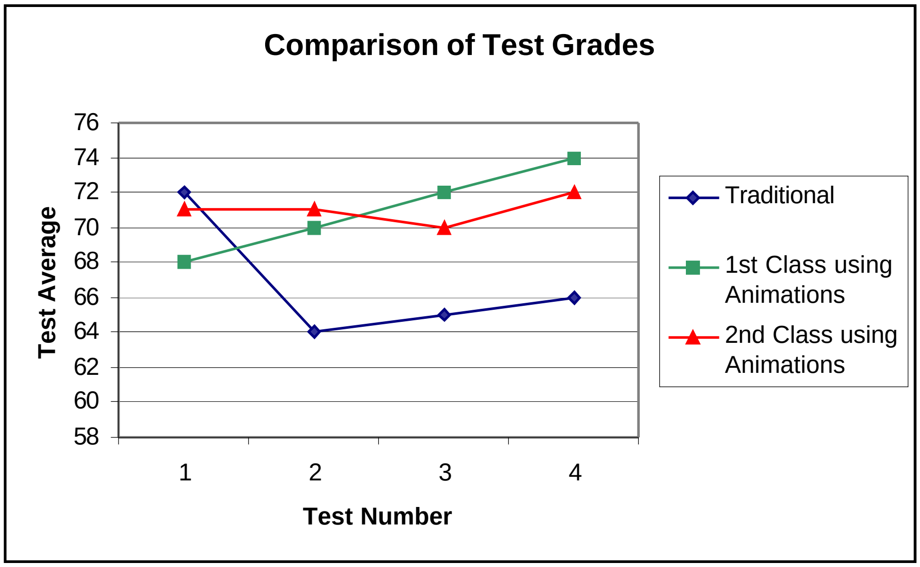


Figure 3: Test Results

Student Opinion Surveys also showed an improvement indicating that students were generally satisfied with these presentation techniques. The graph shown in Figure 5 represents the improvement on two pertinent statements on the student survey. The statements were:

- Statement 1: Instructor has an effective style of presentation

- Statement 2: Instructor makes good use of examples and illustrations

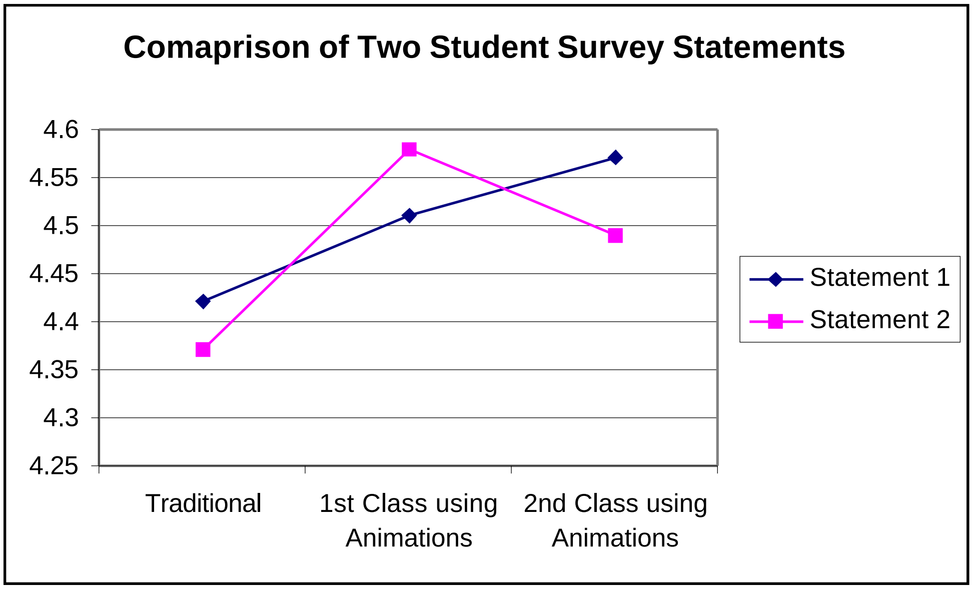


Figure 4: Student Survey Results

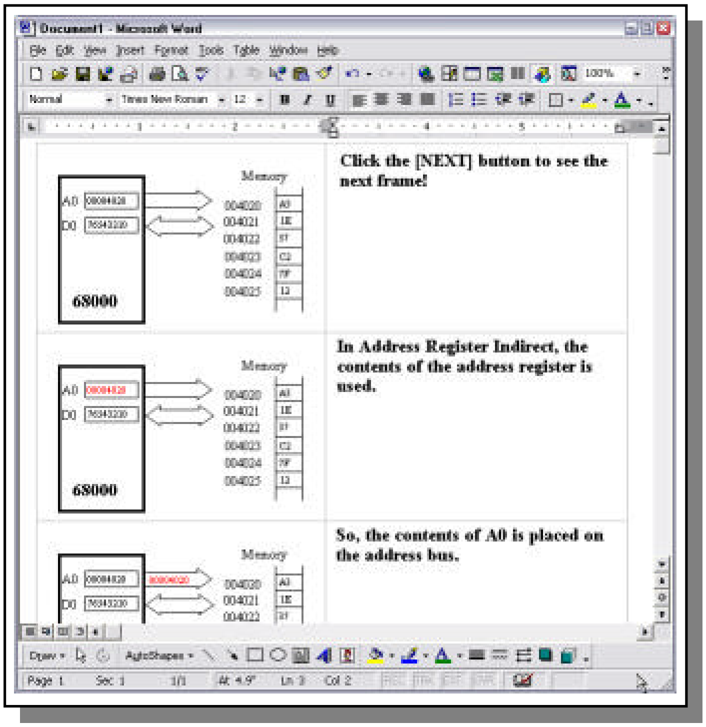
Additionally many students felt compelled to include comments about the animations the first semester they were used. One, which is quoted here, is from a student in a different class: "I saw Steve using an animation in his micro course the other day. When is he going to do this for my class? It looked great."

The results were so positive that a complete web site and several animations were created for the remainder of the course materials. This produced an unexpected and much desired outcome: students began printing and reading the written information from the web site. They were better prepared for class because they read the materials, highlighted important points within the text, and asked better questions during the lecture periods.

Technique:

The technique used to create the animations was simple and made use of available software (no expensive, time intensive software was used). Of course, more detailed and rich images can be created and used for the animations, but is not necessary in most cases. Additionally, if time and money are in short supply, the technique shown below works quite well.

The images were created in Microsoft Word's Picture Editor and placed in a table as shown below:



Click the [NEXT] button to see the next frame!

In Address Register Indirect, the contents of the address register is used.

So, the contents of A0 is placed on the address bus.

Figure 5: Using Word to create animations

The first image was created as a line drawing. To the right, a second image, containing a textual explanation of the current step being shown was created. It is important to note that the image containing the textual information must be large enough to contain the largest explanation in the series of images used in the animation. This is because the JavaScript program used to animate the images, defines an area of specific size for each of the two images in the animation. Both Netscape and Internet Explorer will scale the images to fit in this area. If the actual images are different sizes, you will get some unexpected (and unreadable) results.

The two images were then copied to the next row of the table. The second set of images was edited so that it was identical in size and placement to the first set (no jitter in the animation) and contained only one change. This process continued until all of the images were created. The Word document was then saved as a Web Document, which causes Word to convert each of the drawings into gif images.

Once the images (in gif format) were saved and properly named, the JavaScript file was created. Figure 6 shows the JavaScript code for a three-image animation. The images were named ***1.gif, ***2.gif, etc. The explanations (also in gif format) were named ***1m.gif, ***2m.gif, etc.

```
<script LANGUAGE="JavaScript">
<!-- Begin to hide script contents from old browsers.

        var image = new Array()
        var message = new Array()

        image[1] = new Image();    // image array cell definitions
        image[1].src = "aridn1.gif"
        image[2] = new Image();
        image[2].src = "aridn2.gif"
        image[3] = new Image();
        image[3].src = "aridn3.gif"

        message[1] = new Image();    //message (explanations) array cell definitions
        message[1].src = "aridn1m.gif"
        message[2] = new Image();
        message[2].src = "aridn2m.gif"
        message[3] = new Image();
        message[3].src = "aridn3m.gif"

        var imgNum = 1;

function nextImg() {                    //function to display the Next Image in the animation
        ++imgNum;                       //when the NEXT button is clicked
                if (imgNum>3) imgNum=1;
                document ["picture"].src = image[imgNum].src;
                document ["message"].src = message[imgNum].src;  }

function prevImg(){                     //function to display the Previous Image in the animation
        --imgNum;                       //when the PREVIOUS button is clicked
                if (imgNum<1) imgNum=3;
                document ["picture"].src = image[imgNum].src;
                document ["message"].src = message[imgNum].src;  }

// End hiding -->
</script>
```

Figure 6: JavaScript Code for Animation

The first two lines define new arrays. Immediately following the array definitions is a section of code that populates each cell in the image array and the explanation text (message array). The function *nextImg()* first increments the variable *imgNum* and then checks to see if this is the last image in the sequence. If it is the last image, the function assigns *imgNum* a value of one so that the first images will be redisplayed. The function concludes by placing the two images into the appropriate areas on the web page (see HTML code below). The *prevImg()* function performs the exact opposite operations as the *nextImg()* function.

The html code used to create the **Next** and **Previous** buttons for navigation and to display the first image and explanation is shown below.

```
<body bgcolor="white" TOPMARGIN="0" LEFTMARGIN="0" LINK="#000000" LINK="#000000">
<h3><b><center>Addressing Mode: Address Register Indirect </center></h3></b>
<table WIDTH="600">
<td WIDTH="300"><img SRC="aridn1.gif" NAME="picture" WIDTH="276" HEIGHT="190">
<td WIDTH="300" VAlign="top">
<h3><b><center>Instruction:  MOVE.W D0,(A0)</center></h3></b><br>
<img SRC="aridn1m.gif" NAME="message" WIDTH="288" HEIGHT="72">
</table>
<p>
<form>
<table WIDTH="200" BORDER="0">
<td width="20" Valign="top" Align="left">
<td width="80" Valign="top" Align="left">
<input Type="button" VALUE="Previous" onClick="prevImg(1)"><br>
<td width="80" Valign="top" Align="left">
<input Type="button" VALUE="Next" onClick="nextImg(1)"><br>
</table>
</form>
<p>
```

Figure 7: HTML Code for Animation

The first two lines setup background and link colors and show a title centered on the page. The third line defines a table of 600-pixel width. Line 4 creates the first column of this table, loads the first image (***1.gif) and names this cell “picture”. This links back to the JavaScript code and the *nextImg()* function. The *nextImg()* function specifies the image to be loaded into “picture”. Lastly, this line defines the area for these images. The sixth line defines the same parameters for the explanations. The remainder of the HTML file sets up a table for the placement of the NEXT and Previous buttons and calls the appropriate JavaScript function when clicked.

Conclusion:

The use of animations will make teaching and learning difficult topics more efficient and effective. Because the method of using animations addresses many learning style, students achieved a better understanding of complex topics earlier in the course as is evidenced in Figure 3. This led to better student performance on projects and assignments. Because less time was used in discussing addressing modes and students showed better understanding earlier in the semester, the course, with the use of these animations, became more effective and more efficient.

Students began to show more interest in the course. When students discovered that these course materials were available on the web, they began visiting the web site, printing, reading and highlighting the information provided, and bringing the printouts to class, indicating an interest and preparedness not evident before the animations were developed. Students were better prepared. It should be noted here that no assignments were given to the students to visit, print and read these course materials. Students were becoming interested enough to do so on their own.

Simple line drawings are typically all that is needed, thereby sparing faculty the time and cost of using expensive software products. The software used to develop the animations was Microsoft Word, and a text editor, usually present on any faculty member's office computer. The techniques shown here may be copied and, with little editing, altered to fit your needs.

There are other methods that could be used to achieve the same results. A sequence of Power Point slides can be used in a lecture to animate a sequence of images. This does not work as well when placed on the Internet because the images are large and take time to download.

Macromedia's Flash is an excellent tool to create animations. The author has used it several times when audio files needed to be synchronized with the animations. Although the author would recommend using products such as Flash for animation creation, it does have a few disadvantages:

1. Cost: although not real expensive, it is probably not currently on your office computer,
2. Time: it requires more time to develop and synchronize the audio and image files and additional time to learn to use Flash effectively,
3. File Size: the resulting file size is usually fairly large,
4. Plug-ins: The use of Flash animations requires that the students download the plug-in.

Bibliography:

[1]. Felder, Richard, "Reaching the Second Tier: Learning and Teaching Styles in College Science Education", J. College Science Teaching, 23(5), 286-290 (1993)
[2]. Haque, Mohammed E., "Web-based Visualization Techniques for Structural Design Education", Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition

Biographical Information:

Stephen J Kuyath is an Assistant Professor of Engineering Technology at the University of North Carolina at Charlotte. He has taught engineering technology courses at the college level for over 18 years. His interest of online and distance education began in 1995 while teaching at Central Piedmont Community College in Charlotte, NC.