
AC 2011-457: A COMPARISON OF HANDS-ON VERSUS REMOTE LABORATORY EXPERIENCE FOR INTRODUCTORY MICROPROCESSORS COURSES

Brock J. LaMeres, Montana State University

Brock J. LaMeres is an Assistant Professor in the electrical and computer engineering department at Montana State University (MSU). LaMeres teaches and conducts research in the area of digital systems and engineering education. LaMeres is currently studying the effectiveness of online delivery of engineering education including the impact of remote laboratory experiences. LaMeres' research group is also studying the effective hardware/software partitioning using reprogrammable fabrics. This work involves exploiting the flexibility of modern FPGAs to optimize the performance of a digital system depending on the application need (i.e., performance, power, size, or fault tolerance). LaMeres' research is sponsored by NASA, the National Science Foundation, the Montana Space Grant Consortium, the National Space Grant Consortium, and the Office of Naval Research.

Carolyn Plumb, Montana State University

Carolyn Plumb is the Director of Educational Innovation and Strategic Projects at Montana State University. She has been involved in engineering education for over 20 years.

A Comparison of Hands-On versus Remote Laboratory Experience for Introductory Microprocessors Courses

Abstract

This paper describes an approach to assessing and improving the understanding of microprocessor systems for electrical and computer engineering students by developing measurement-based laboratory experiments. During fall semester of 2009, we assessed the level of understanding of microprocessor systems on a control group using five learning objectives. Students in the control group were enrolled in EE 371, Microprocessor Hardware and Software Systems, a required course in the electrical engineering and computer engineering programs. We measured the level of understanding using a set of assessment tools that includes self surveys, weighted multiple choice questions, and short answer questions. These assessments set a baseline measure on the five learning objectives for our current microprocessor curriculum. In fall of 2010, we introduced measurement-based laboratory experiments using logic analyzers. The measurement-based experiments were introduced in two forms: hands-on and remote operation. Assessment data was collected for both experimental groups and compared to the control group from fall 2009 to determine (1) if the level of understanding of microprocessor systems is improved by adding hands-on measurements and (2) if a remote laboratory experience can maintain or improve the level of understanding compared to the control group.

In this paper, we summarize the development of the assessment tools used in this project, including the creation of a grading rubric to achieve a finer resolution on the scores of the short answer questions. We also report on the comparisons between the three groups (control group, hands-on measurement group, and remote measurement group) in regard to self-reported learning on the five learning objectives from survey results. Finally, we present comparisons of direct measurement of learning on the five learning objectives from multiple choice and short answer quiz results.

1. Introduction

Introductory microprocessor hardware and software courses are present in nearly all ABET accredited electrical and computer engineering (ECE) undergraduate curriculums. These courses, typically taught in the junior year, expose students to the basic architecture of a microcomputer including microprocessors, memory, and input/output (IO) systems. A common approach to teaching microcomputers is to use assembly language to program the computer to perform basic tasks. This approach shows the students how the computer operates at the individual instruction level and how data is transferred between sub-systems. A standard computer platform (most commonly a microcontroller) is typically used and its instruction set is covered to show what operations are necessary in a microcomputer. While learning assembly language and a processor-specific instruction set is not the primary learning objective, it has been shown to be an effective way to introduce the makeup of a computer.

At Montana State University, juniors in the undergraduate ECE curriculum take a course titled “EE371 – Introduction to Microprocessor HW/SW”. This course uses the FreeScale HCS12 microprocessor incorporated in an MC9S12C128 single-chip microcontroller that contains all of

the necessary random access memory (RAM) non-volatile memory (ROM), and IO circuitry. This course is 4 credits and is taught on a 16-week semester schedule. The 3-credit lecture portion of the course meets 3 times a week for 50 minutes each. The 1-credit lab portion meets once a week for 2 hours. Lab exercises have historically consisted of students writing assembly language programs to accomplish a task with the microcomputer. Basic IO (buttons, LEDs, buzzers) provide the human interface to the computer. The FreeScale CodeWarrior Development system is used to create the programs, download to the microcontroller, and provide a debugging environment.

In 2009, we began a project to investigate whether the understanding of a microprocessor could be improved if measurement was included in the laboratory experience. Until this project, students did not have any measurement visibility into the computer system. Only the debugger and LEDs were used to see the step-by-step operation of the computer. A logic analyzer was selected as the piece of test equipment to introduce measurement into the labs. A logic analyzer observes a large number of digital signals (typically >16) and provides the data to the user in either a waveform or listing format. Each of the labs in the EE371 course was redesigned to have a measurement component to see if the student understanding of the computer was improved.

An additional benefit to using a logic analyzer is that the instrument itself can be used to program the microcontroller. The instrument is simply a Windows-based workstation with additional measurement hardware and software to perform the logic analysis. The workstation itself contains all of the features that a standard PC does including remote access capability, a USB interface, and the ability to run Windows applications. In this project, we used the logic analyzer as the lab computer. CodeWarrior was installed on the logic analyzer and the students were able to develop programs, download them to the microcontroller, and take measurements on a single platform. By creating the lab setup in this way, the *Remote Desktop Connection* (RDC) feature of the Windows operating system can be used to perform the entire lab remotely. By including a web cam in front of the microcontroller platform, the LED IO can be observed. Also, the *Virtual Component* feature within the CodeWarrior system allows signals to be sent to the microcontroller during operation to mimic button presses. As each of the EE371 labs was redesigned for this project, being able to remotely perform the labs was a requirement. This capability enabled this project to study not only if measurement capability improved understanding, but how an equivalent *remote access* lab experience compared.

In order to gauge whether the students' understanding of the microcomputer was improved, 5 learning outcomes were selected for this project:

- 1) Describe the basic architecture of a stored-program computer;
- 2) Describe the addressing modes of a microprocessor;
- 3) Describe a typical I/O interface and understand its timing;
- 4) Analyze a timing diagram of the interaction between the microprocessor and memory;
- 5) Synthesize a timing diagram of a given READ/WRITE cycle between the microprocessor and memory.

For each of these five outcomes, assessment tools were created, including self evaluation surveys, weighted multiple choice (MC) questions, and short answer questions (SA). The assessment data was collected using the quiz feature within the *Desire2Learn* web-based course management system. None of the assessment data from these assessment tools was used in computing the students' grades for the course. These assessment tools were used to collect data on the Fall 2009 EE371 students, who did not have access to the logic analyzers in lab. This class served as the *control group* for the project. In the Fall 2010 semester, the EE371 class was divided into two experiment groups. The first group performed all of the labs hands-on using the newly designed lab assignments that included measurement with the logic analyzer. The second group performed all of the labs remotely using the Remote Desktop Connection to the logic analyzer. The assessment scores for each of the three groups were compared for each of the 5 outcomes to see how the inclusion of measurement impacted student understanding.

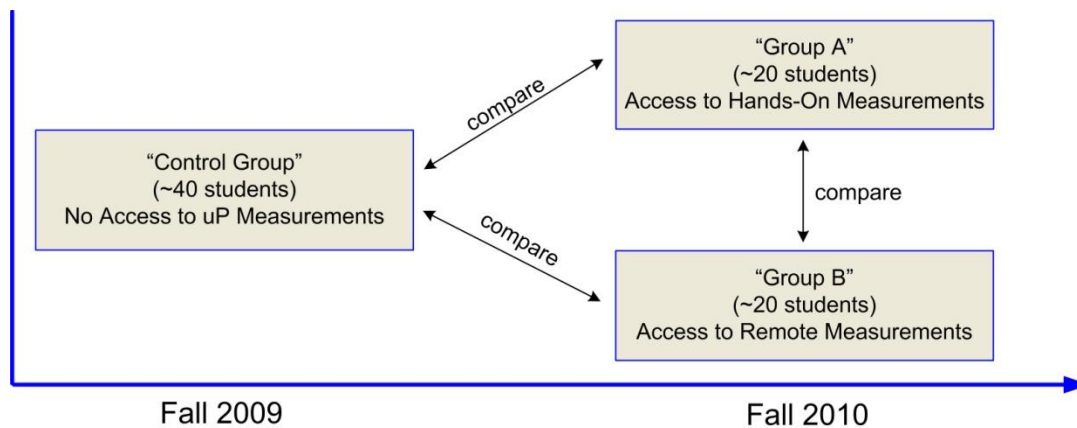


Figure 1. Assessment timeline for control and experimental groups in this project.

2. Lab Development

In EE371, there were 11 lab assignments. The assessment tools were used to collect data after particular labs that covered some aspect of the 5 learning outcomes. The 11 lab topics were the same in the Fall 2009 control group as they were for the Fall 2010 experimental groups. The only difference was the inclusion of measurement and any modifications to the assigned task to facilitate the measurement.

Lab 1 – Introduction to uP HW/SW: The objective of this lab was to introduce the students to the lab setup. The students were given an example source code file to make a repeating pattern on the LEDs and a step-by-step tutorial on how to run the CodeWarrior software to assemble, download, and run the program. The students were also given instructions on how to setup the logic analyzer and take a measurement. At the end this lab, the students were able to see the flashing pattern on the LEDs and also see the data in the logic analyzer. The following figure shows the logic analyzer measurement students obtained at the end of the lab which shows the data on 4 LEDs on the microcontroller board.

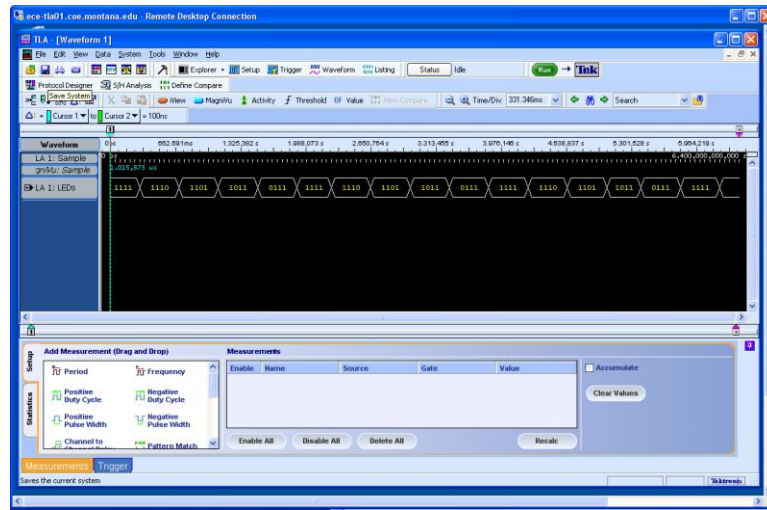


Figure 2. Logic analyzer measurement for lab #1.

Lab 2 – LED Parallel I/O: The objective of this lab was to have the students design a program to output a 4-bit counting pattern on the IO of the microcontroller. The logic analyzer was used to observe the pattern to verify functional operation on the IO port in addition to measure the period of the counter. The following figure shows the logic analyzer measurement students obtained at the end of the lab.

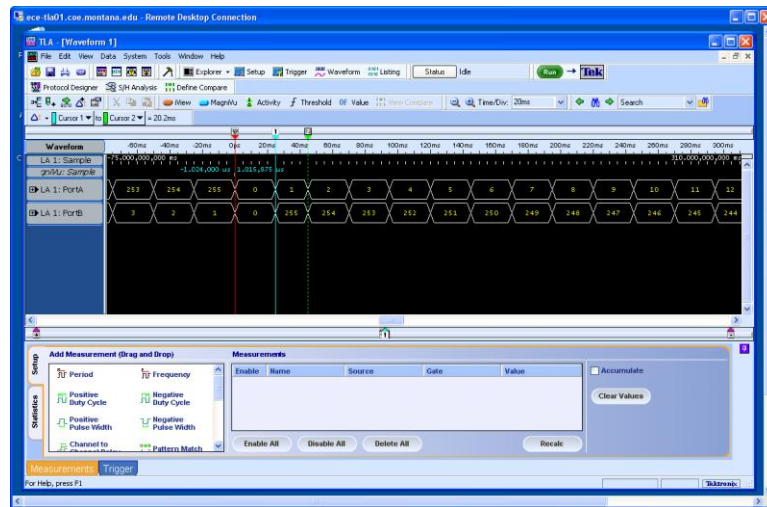


Figure 3. Logic analyzer measurement for lab #2.

Lab 3 – Addressing Modes: The objective of this lab was to have the students design a program to retrieve data from memory using the different addressing modes that are available in the microprocessor. Each time the data was retrieved from memory, it was written to the IO port of the microcontroller for observation with the logic analyzer. This allowed to the students to see the data retrieval in real-time.

Lab 4 – HEX to ASCII: The objective of this lab was to have the students design a program to convert nibbles of HEX data into an ASCII code. A table of HEX data was provided that the students needed to operate on. After conversion, the ASCII codes were written to the IO port for observation with the logic analyzer. The logic analyzer is able to display the binary codes as ASCII symbols so the students were able to see the actual character that the HEX data represented.

Lab 5 – Addressing the STACK: The objective of this lab was to expose students to the STACK and how it is used to hold temporary variables. A table of data was provided to the students. The students needed to design a program to use the STACK to continually reverse the contents of the table using the STACK. As the STACK was accessed, the Stack Pointer Address Register was written to the IO port for observation with the logic analyzer. This allowed the students to observe how the STACK was accessed during program operation. The following figure shows the logic analyzer measurement students obtained at the end of the lab.

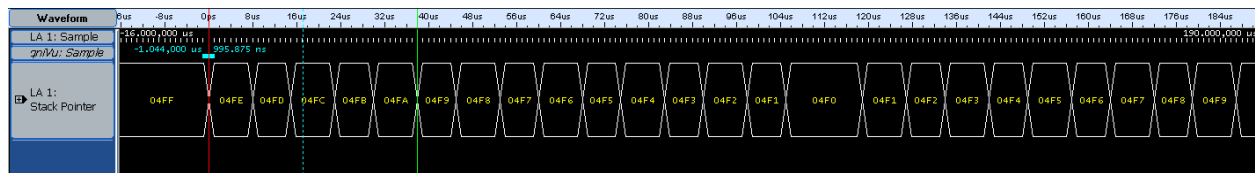


Figure 4. Logic analyzer measurement for lab #5.

Lab 6 – Reading and Writing External IO: The objective of this lab was to expose students to the concept of *polling* an IO pin and using the external event to trigger code execution. The students designed a program to continually poll a button that was connected to the microcontroller. When the button was pressed, their program would output a pattern to the LEDs and then return to polling. The logic analyzer was used to measure the pattern written to the LEDs. The students needed to setup the logic analyzer measurement to only acquire data once the button was pressed.

Lab 7 – Instruction Speed & External IO: The first objective of this lab was to use the logic analyzer to measure the individual instruction speed of various instructions in of the microprocessor. The students first calculated the expected execution time by hand and then verified their results by observing the instruction results as they were written to the IO. The second objective of the lab was to expose the students to how the external IO of the microcontroller can be used to interface with an external memory device. The students needed to design a program to output a set of control lines to perform a memory WRITE following a provided protocol. This not only exposed the students to interfacing techniques but also reinforced the steps necessary to perform a WRITE operation. The following figure shows the logic analyzer measurement students obtained at the end of the lab.

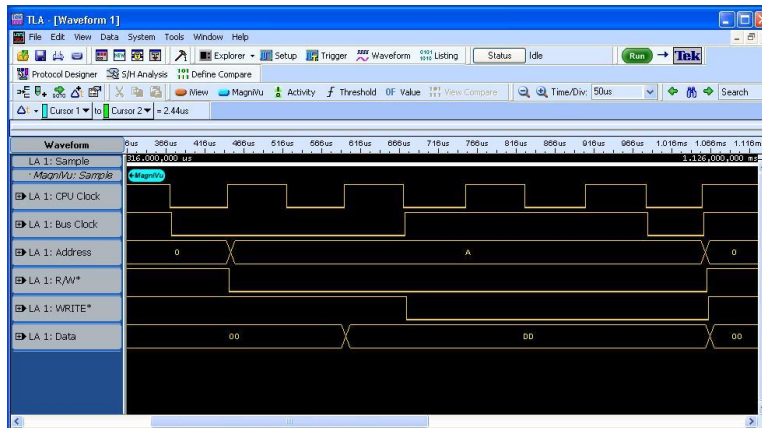


Figure 5. Logic analyzer measurement for lab #8.

Lab 8 – Interrupts: The objective of this lab was to expose students to interrupts in a computer system. The students were to write a program that would generate an interrupt when a button was pressed. The interrupt would output a pattern to the LEDs. During the interrupt service routine, the students were to write the contents of the STACK to the IO for observation with the logic analyzer. This illustrated how the STACK was used to preserve register contents during an interrupt.

Lab 9 – Multiple Interrupts: The objective of this lab was to expose students to multiple interrupts and the concept of priority in a computer system. The students were to write a program that would handle interrupts being generated by multiple buttons. The program prioritized the service of each input. The logic analyzer was used to again observe the contents of the STACK during an interrupt service routine and helped the students visualize what happens when a higher priority event interrupts a lower priority event.

Lab 10 – Timers: The objective of this lab was to expose students to using the timer system on a microcontroller to generate events at deterministic periods of time. The students were to write a program to output a counter pattern to the IO at a given frequency. The logic analyzer was used to measure how close their frequency was to the design objective.

Lab 11 – Multiple Timers: The objective of this lab was to expose students to using multiple timer systems on a microcontroller to trigger multiple events. The students wrote a program to toggle a signal that was connected to a buzzer. The frequency of the signal was controlled by a timer and could be changed to alter the tone of the buzzer. The frequency of the signal was measured with the logic analyzer. A second timer was used to update a pattern on the LED at a different frequency and again its speed was measured with the logic analyzer.

Each of these labs could be performed either hands-on or remotely. The hands-on group performed the labs at a workspace that contained the logic analyzer and the FreeScale microcontroller board. The remote lab stations were kept in a quarantined room that was only accessed by the lab instructor in order to keep the hardware in working order. The remote group accessed these stations using Remote Desktop Connection from the same room that the hands-on group was working so they had access to help from the lab instructor. The logic analyzer

connection was pre-defined and the students were not allowed to change it. This allowed both the hands-on and remote groups to have the same experience. This is typically how logic analyzers are used.



Figure 6. Laboratory setup for the hands-on (left) and remote (right) users.

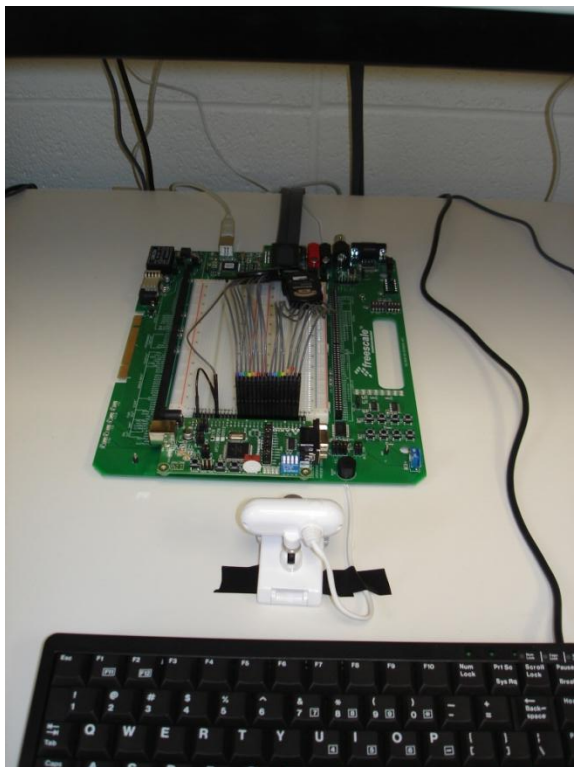


Figure 7. Logic analyzer connection to the FreeScale microcontroller board. Also shown is the web cam used for LED observation by the remote users.

3. Assessment Tool Development

The assessment data was collected using the quiz feature within the *Desire2Learn* course management system, which allowed auto grading of the survey and multiple choice questions. Online quizzes were given after exercises 3-9 to collect information on the student understanding of the learning outcomes. The following table gives the topic and order of the 11 laboratory exercises conducted. The table shows, for each lab session, the assessment tool used and the targeted outcome(s) measured [7].

Table 1. List of laboratory experiments, targeted outcomes, and assessment tools used.

Laboratory Experiment	Outcome(s)	Assessment Tool(s)
1 – Introduction to uP HW/SW	-	-
2 – LED Parallel I/O	-	-
3 – Addressing Modes	1,2,3,4,5	Self Evaluation Pre-Survey
4 – Addressing Tables	2	1 Multiple Choice (MC) 1 Short Answer (SA)
5 – Addressing the STACK	1	4 Multiple Choice (MC) 2 Short Answer (SA)
	2	2 Multiple Choice (MC)
6 – Reading/Writing External I/O	2	1 Multiple Choice (MC) 1 Short Answer (SA)
	3	6 Multiple Choice (MC) 1 Short Answer (SA)
7 – External Interrupts	3	1 Multiple Choice (MC) 2 Short Answer (SA)
	5	5 Multiple Choice (MC)
8 – Multiple Interrupts	4	2 Multiple Choice (MC) 2 Short Answer (SA)
	5	2 Short Answer (SA)
9 - Timers	1, 2, 3, 4, 5	Self Evaluation Post-Survey
10 - Timers	-	-
11 – A/D Converter	-	-

Multiple measures of each outcome were used in order to overcome a statistically small sample size (14-40 students) and also to validate our results. We developed at least two multiple choice (MC) questions and at least two short answer (SA) questions for each of the 5 outcomes. A self evaluation question for each of the 5 outcomes was given twice during the semester, once as a pre-test before performing the relevant lab exercises and once as a post-test after. The following table lists the 5 outcomes, as well as the number and type of assessment questions for each.

Table 2. Number of Weighted Multiple Choice (MC) and Short Answer (SA) Questions Per Outcome

Outcome	Multiple Choice	Short Answer
1. Describe the basic architecture of a stored program computer	4	2
2. Describe the addressing modes used in a microprocessor	4	2
3. Describe the functional operation and timing dependencies of a microprocessor I/O interface	7	3
4. Analyze a timing diagram of the interaction between a microprocessor & memory	2	2
5. Synthesize a timing diagram of the interaction between a microprocessor & memory	5	2

3.1 Self Evaluation Surveys

A survey measuring student self perception of knowledge of the project objectives was developed, and students completed the survey in *Desire2Learn* during the third week of the course (after Lab 3) as well as at the end of the course. The survey asked students to evaluate, on a scale from 0 to 10, their understanding of each of the five objectives. The following table lists the self evaluation survey question for each objective [7].

Table 3. Self Evaluation Survey Questions Per Objective.

Survey Question	Objective
Based on everything you have learned in this course so far, evaluate on a scale from 0-10 your own ability to describe the basic architecture of a stored program computer:	1
Based on everything you have learned in this course so far, evaluate on a scale from 0-10 your own ability to describe the addressing modes used in a microprocessor:	2
Based on everything you have learned in this course so far, evaluate on a scale from 0-10 your own ability to describe the functional operation and timing dependencies of a microprocessor I/O interface:	3
Based on everything you have learned in this course so far, evaluate on a scale from 0-10 your own ability to analyze a timing diagram of the interaction between a microprocessor & memory:	4
Based on everything you have learned in this course so far, evaluate on a scale from 0-10 your own ability to synthesize a timing diagram of the interaction between a microprocessor & memory:	5

3.2 Multiple Choice (MC) Questions

We developed multiple choice questions targeting each objective. Our rationale for using multiple choice questions was two-fold. First, we wanted questions that would require some thought and actual knowledge of the subject, and we wished these to give us an accurate measure of the students' level of understanding of an objective. Second, by carefully designing the questions, the correct answers, and the distracters, we could more accurately evaluate student performance changes from one year to the next. We accomplished this by having answers that

have weighted scores (i.e., 0 through 5 with 5 being the highest) instead of simply being right or wrong. For each of the multiple choice questions, the answers provided included one answer that was the *best* choice representing the *most correct* answer. This answer was assigned the most points (i.e., 5). Within the answer list were other choices (distracters) which were the 2nd *best choice*, 3rd *best choice*, etc., each with fewer points assigned. In creating this kind of multiple choice question, the best answer is generated first. The next choices are generated by making them about the same length, with the same grammatical structure, and plausible but with less detail, perhaps, than the best choice. This strategy gives us a grading scale that helps us understand not only how many students are giving the correct answers but also the degree to which students understand the topic. Students were required to read through each of the answers of the question in detail and choose which they felt was the most accurate response. An example of a multiple choice question implementing the grading rubric is given below [7]:

Multiple Choice Question Example (Objective #1)

Which of the following most completely describes the function of the stack in a microcontroller?

- A The stack is in ROM and is used to access constant data used in your programs.
(0 points)

- B The stack is in RAM and is used to store temporary variable data and subroutine return addresses using indexed addressing modes.
(1 points)

- C The stack allows you to have nested subroutines in your programs.
(2 points)

- D The stack is used to store registers but you have to initialize the stack pointer register first.
(3 points)

- E The stack is in RAM and is used to store temporary variable data and subroutine return addresses.
(5 points)

The answers were randomized in the *Desire2Learn* quiz system so that each student was given the answers in a different order during the quiz to remove any dependency between the objective score and the response location within the quiz.

4. Outcome Results

The pre-survey and post-survey averages (on a scale from 0 to 10) are shown below:

Table 4. Self Evaluation Survey Results Per Objective.

	Group 1 (Control)		Group 2 (Hands-On)		Group 3 (Remote)	
	Pre N=46	Post n=41	Pre n=32	Post n=30	Pre n=15	Post n=14
Outcome 1 Question	6.304	7.293	6.563	7.600	5.267	6.143
Outcome 2 Question	6.891	8.634	6.625	8.300	5.600	7.786
Outcome 3 Question	5.283	6.683	5.844	6.767	3.867	5.429
Outcome 4 Question	6.152	7.512	6.688	7.767	5.067	7.071
Outcome 5 Question	5.587	6.805	6.031	7.300	4.400	6.429

Using SPSS, we did an analysis of covariance (ANCOVA) on the survey data. The ANCOVA used the pre-survey data as a covariate to determine whether or not there was a difference among the three groups in the post-survey data, taking into account the students' perception of their knowledge of the learning outcomes early in the course. Prior to the ANCOVA, we tested the homogeneity of slopes of the pre-test to post-test scores of the 3 groups and found a significance only for question 5. The ANCOVA showed no significant differences among the three groups that could be attributed to the group. For all 5 questions, the differences among the three groups on the post-survey were predicted by the initial responses on the pre-survey. Thus, in regard to the students' self-perception, the two experimental groups did not perceive any greater gains in learning on the outcomes than the control group; however, the remote group perceived the same level of gains in learning as the other two groups

It is possible that the finding of no significant difference among the groups in the survey data could be due to the small number of subjects, particularly in Group B, the remote group. To better understand the effect of the measurement lab, both in the hands-on condition and the remote condition, we turned to some of our other assessment data.

The following 5 figures show the results of each assessment tool for each of the outcomes. These scores are normalized to a 100-point scale. The survey data is presented in terms of the percent increase between the pre- and post- surveys. In all cases the post-survey scores were higher.

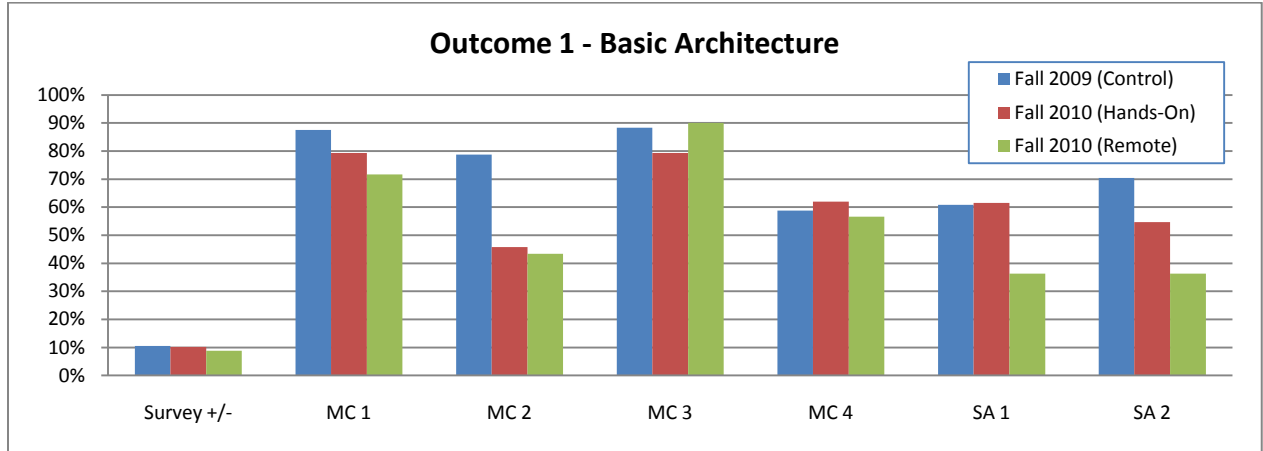


Figure 8. Experiment results for outcome 1. (MC = Multiple Choice; SA = Short Answer)

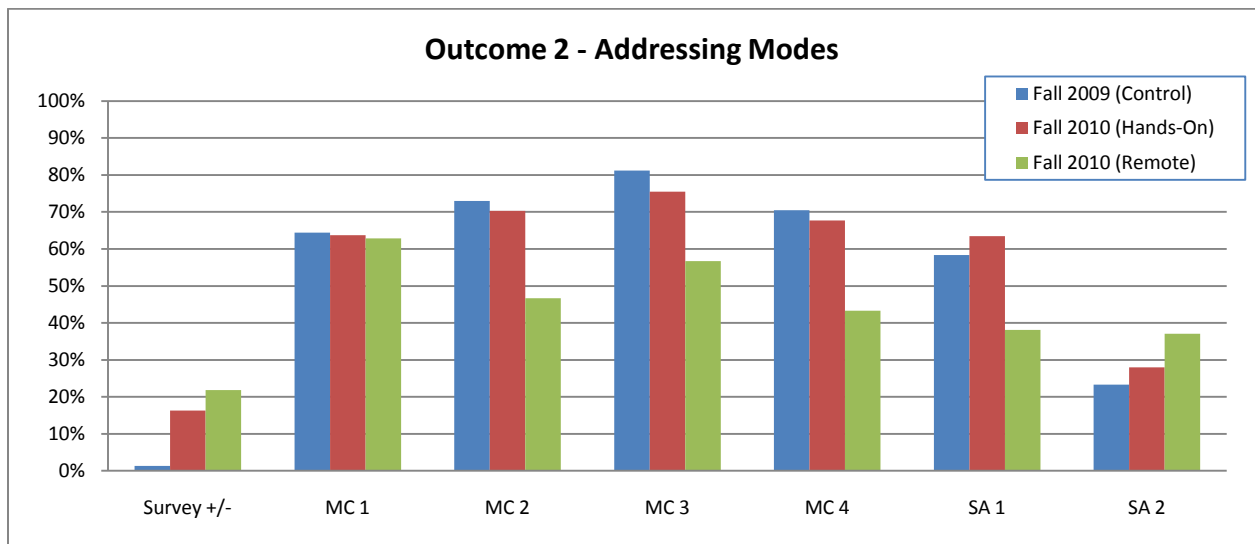


Figure 9. Experiment results for outcome 2. (MC = Multiple Choice; SA = Short Answer)

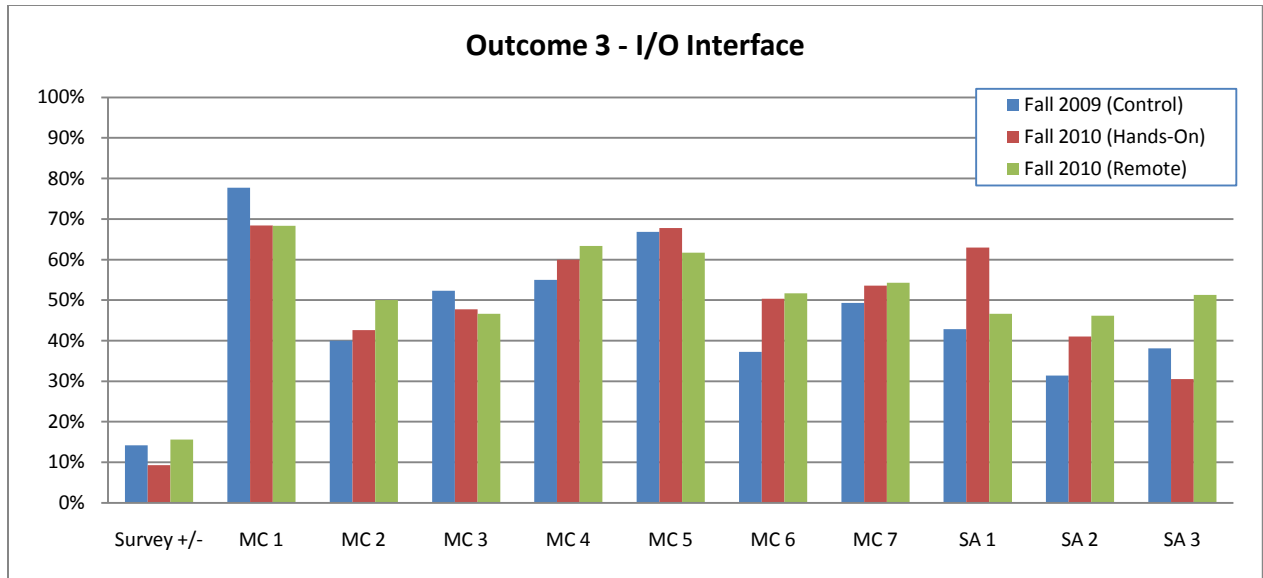


Figure 10. Experiment results for outcome 3. (MC = Multiple Choice; SA = Short Answer)

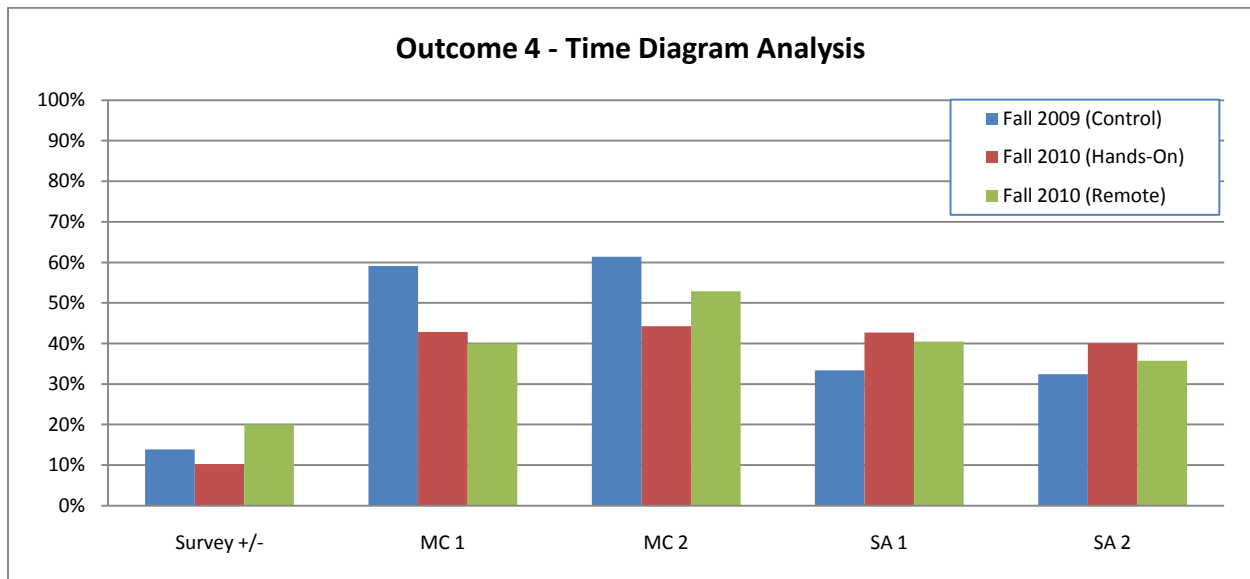


Figure 11. Experiment results for outcome 4. (MC = Multiple Choice; SA = Short Answer)

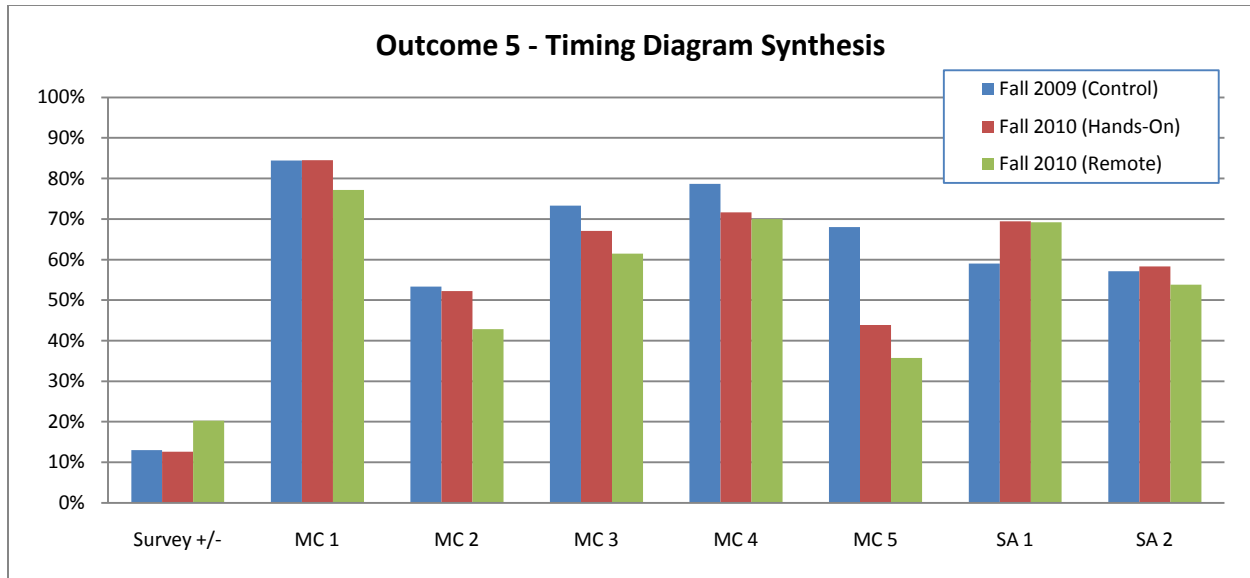


Figure 12. Experiment results for outcome 5. (MC = Multiple Choice; SA = Short Answer)

These results show that for some of the assessment questions there was an increase in student understanding and for others there was not. Detailed examination of the results indicate that the inclusion of measurement showed an increase in understanding for topics that require greater visualization such as input/output operations and timing diagrams. Measurement did not have a major impact on the topics that are not typically visual such as architecture and addressing modes. One takeaway from these results is that the remote experience was not significantly different from the control or hands-on. This means that our approach could provide a meaningful laboratory experience if this course were offered remotely.

5. Summary

In summary, this project looked at if including a measurement experience in a junior-level microprocessor course could improve students' understanding of a microcomputer. It further studied if the measurement-based laboratory exercises could be performed remotely using a particular lab setup and provide the same level of lab experience. We presented the design of the laboratory experience and the development of the assessment tools used in our project. The results of three groups of students participating in this project are presented. It was discovered that measurement aided in the student's understanding of visual topics but did not have significant affect on non-visual topics. It was also found that the remote laboratory experience did not have significantly different outcome scores compared to the control or hands-on group.

6. Acknowledgements

The authors would like to thank the National Science Foundation for funding this project through the Course, Curriculum and Laboratory Improvement (CCLI) Program (Award #0836961) under the Division of Undergraduate Education. The authors would also like to Tektronix Inc. for support of this work through equipment donations. The additional equipment provided by Tektronix to augment the equipment procured using funding from the NSF CCLI award allowed full coverage of each lab stations in our digital lab to be outfitted with logic analyzers.

Bibliography

1. W. Wolf, J. Madsen, "Embedded systems education for the future," *Proceedings of the IEEE*, vol. 88, no. 1, pp. 23-30, Jan. 2000.
2. W.A. Stapleton, "Microcomputer fundamentals for embedded systems education," *36th ASEE/IEEE Frontiers in Education Conference*, San Diego, CA, Jan. 2006.
3. S. Marín, F. García, R. Torres, S. Vázquez, and A. Moreno, "Implementation of a web-based education tool for digital signal processing teaching using the technological acceptance model," *IEEE Transaction on Education*, vol. 48, no. 4, pp. 632-641, Nov. 2005.
4. J.S. Chenard, A.U. Khalid, M. Prokic, R. Zhang, K.L. Lim, A. Chattopadhyay and Z. Zilic, "Expandable and robust laboratory for microprocessor systems," *IEEE International Conference on Microelectronics Systems Education (MSE'05)*, pp. 65-88, June. 2005.
5. L. Gomes, A. Costa, "Remote laboratory support for an introductory microprocessor course," *IEEE International Conference on Microelectronics Systems Education (MSE'05)*, pp. 21-22, June. 2005.
6. E.P. Ferlin, V. Pilla Jr., "Driving the learning: microprocessors from theory to practice", *Proceedings of the 5th International Conference on Information Technology Based Higher Education and Training*, pp. 459-462, June. 2004.
7. B. LaMeres, C. Plumb, F. Cady, "Improved Student Learning of Microprocessor Systems Through Hands-On and Online Experience", *American Society for Engineering Education (ASEE) Annual Conference*, June 20-23, 2010, Louisville, KY.