Distributed Laboratories: A Finite State Machine Module

Bonnie Ferri, Jill Auerbach, Hongyi Qu
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250
Corresponding author: bonnie@ece.gatech.edu

Abstract:

This paper presents an experimental module for teaching finite state machine concepts. This module is designed for use in a lecture-based course that does not currently have a lab associated with it. For this module, the students do some pre-class preparation, including a pre-lab where they design the state machine circuitry, then they build the design on a protoboard in class while at their desks. The experimental platforms are low weight and powered by 3-AA batteries for portability. A challenge discussed in this paper is the logistics so that this experiment can be completed in a 50 minute class period. The web support includes an instructional video, a fundamental concepts tutorial, a virtual experiment, help for instructors, and a set of on-line quiz questions typical of standard lecture-based test questions. This module has been used in 11 classes and assessment for this experiment has included 471 students.

Keywords: Distributed lab, active learning, finite state machine

Introduction:

Laboratory experiments are a vital source for active learning in Engineering programs, yet the format is often not well suited for incorporation into lecture-based courses. Lab experiments are generally performed in lab courses in centralized locations. A new extension to the laboratory experience is distributed laboratories, which consist of experiments that can be conducted in a variety of places such as a standard classroom or a dorm room. As such, they can be incorporated into traditional courses or distributed from decentralized locations. Recently, there has been a limited development of some hands-on experimental platforms for engineering students to use in the classroom or to take home (Dua, 2005), (Durfee, 2005), (Hendricks, 2005), and (Litwhiler, 2005), (Ferri, 2008), and (Ferri, 2009).

The Finite State Machine Module described in this paper is one of the modules developed as part of a project funded by the NSF CCLI program to develop labs that can be distributed throughout the curriculum, especially in courses that do not currently have labs. The Teaching Enhancement via Small-Scale Affordable Labs (TESSAL) Center, was established to provide cohesive support for the laboratory modules. All of the educational modules in the TESSAL Center share common features:

- Demonstrate fundamental concepts that are often difficult to comprehend in a lecture setting
- Have portable platforms so that they can be performed in class or at home
- Be affordable for mid-sized lecture courses (30-50 students)
- Have web support to promote wide-spread adoption

The TESSAL website, http://www.ece.gatech.edu/research/tessal/index.html, houses the components of each lab module, which include:

- Experimental platform description
- Tutorial on the fundamental theoretical concepts
- Laboratory instructions for students

- Video or interactive demo to demonstrate operation of the experiment
- Quiz questions that are representative of test questions given in a lecture-based course
- Instructions for instructors to build the platform and to implement the lab

Finite State Machine Module

Digital logic courses incorporate many concepts from binary logic and arithmetic and logic gates through memory units and possibly up through datapaths. One of the difficult transitions for students in the course is the step from combinational logic to sequential circuits since time is now a factor. State machines are difficult for students in that they add a dynamic dimension to the circuit. The output of the circuit depends not only on the inputs, but also on the previous state of the circuit. There are many software simulators of state machines, for example, a MATLAB© toolbox (Mathworks, 2010) and a Java applet (Java, 2010). The real experiment, which can be performed in class, provides students with the opportunity to experience first hand the complex behaviors due to memory in sequential circuits. In using the state machine module, students can benefit from an active learning approach (they are actually doing something) that expounds upon the more difficult concepts covered in the lecture-style class. This module presents the material in a problem-based format, which further engages students in the learning process. The associated web site augments the experience by adding multi-media content that parallels the experiment.

The goal of the State Machine Module is to reinforce state machine concepts by having students design and implement a state machine using simple chips and a protoboard. Since this module is made for use in courses that do not have concurrent labs, it also introduces students to basic physical components. Two objectives are to increase students' understanding of state machine concepts and design, and enhance their interest and enthusiasm in learning the course material.

The general steps of finite state machine design and implementation are as follows:

1) Convert a description of the problem into a state transition diagram
2) Transfer the information from the state transition diagram to a state transition logic table that has inputs consisting of system inputs and current values of the state, S_i, and outputs consisting of system outputs and next state values, NS_i.
3) Design a combinational circuit to implement the logic in the table.
4) Select the chips to implement the combinational circuit and to implement the memory portion of the state machine, for example, D flip flops can be used to implement registers.
5) Draw a pin diagram to illustrate how to wire the chips together to implement the state machine logic.
6) Insert the chips into a protoboard and wire the ground, high voltage, enables, and clock pins.
7) Complete the circuit by making the connections indicated from Step 5).
8) Test the circuit.

This module requires students to perform steps 2)-8) in a classroom setting and includes supporting web-based resources. CMOS chips are used for low power, and a battery pack is used for portability.

The state machine experiment consists of a protoboard, a 3-8 decoder, a battery pack (using 3 AA batteries), dip switches, LEDs, D flip flops, and OR gate chips as shown in Figure 1. The experiment requires students to design a state machine from a state transition diagram. This is similar to standard homework and test problems on the subject. The module requires students to go one step further and build the circuit in class and test its behavior. They input signals through the dip switches and observe outputs via the LEDs. The experiment is followed up by a set of on-line quiz questions that resemble those that might be found on a test in a lecture course on the material.

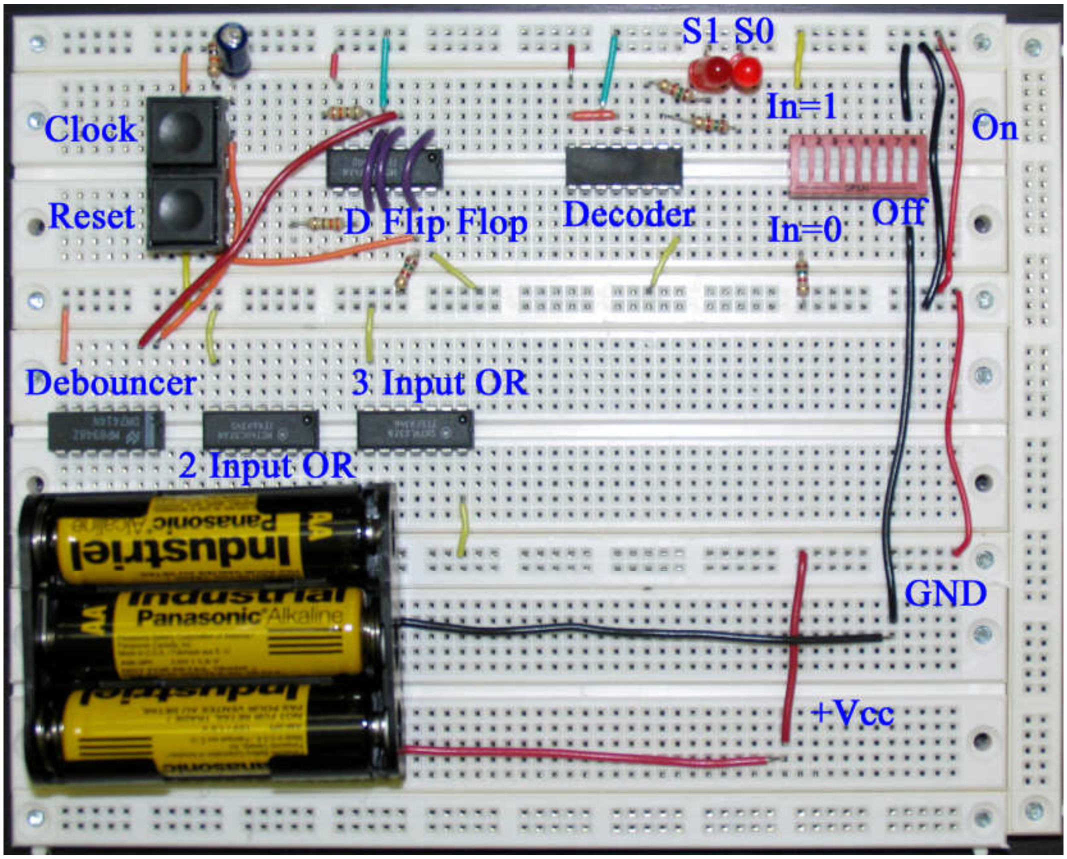


Figure 1: Partially pre-wired protoboard for a state machine implementation.

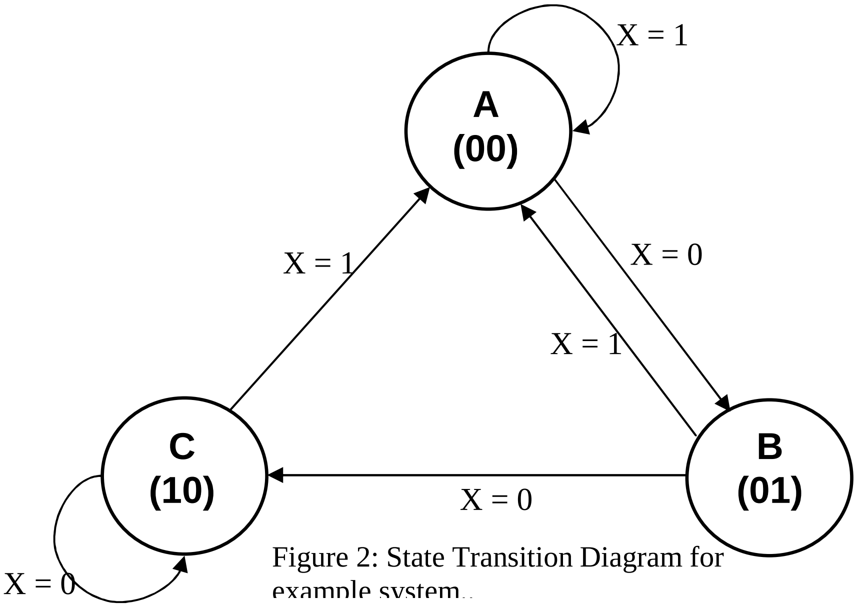


Figure 2: State Transition Diagram for example system..

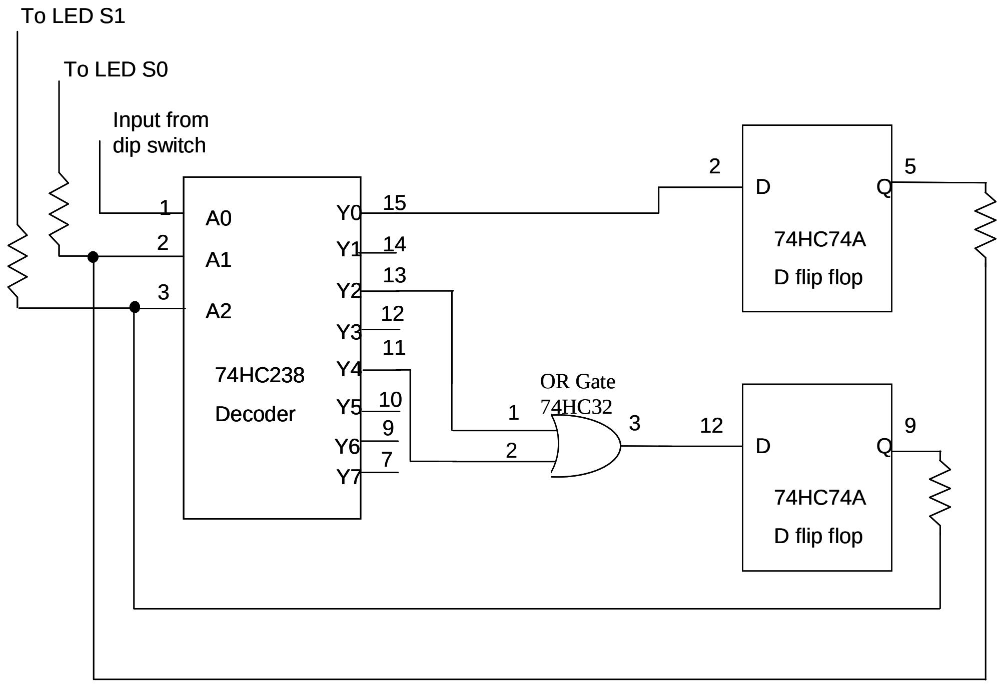


Figure 3: Pin diagram for the example system.

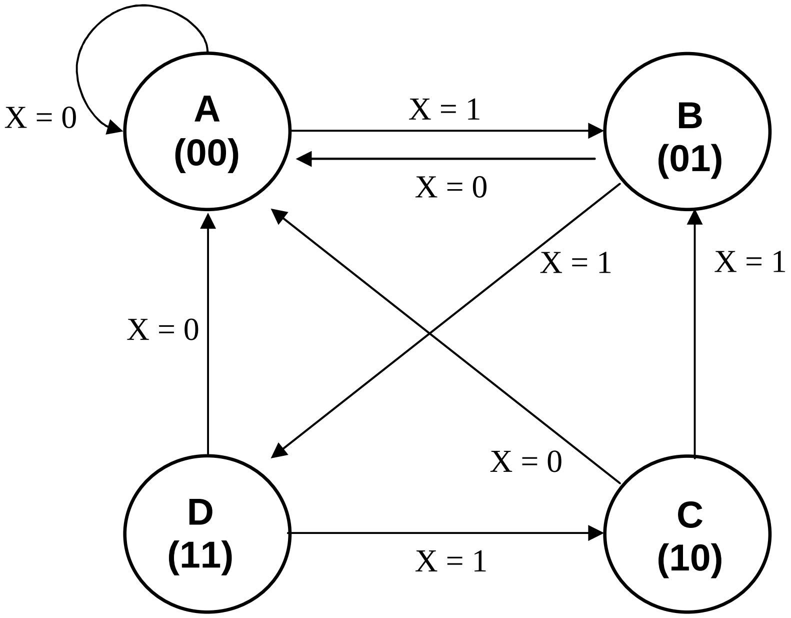


Figure 4: State machine assignment for design and implementation.

In the lab, students are required first to build and test the example finite state machine shown in Figure 2. This machine only requires a 2-input OR gate chip, the decoder, and the flip-flop. The pin diagram for the circuit is given to the students and is shown in Figure 3. The students must use the procedure outlined in the sample state machine to design and implement a different machine, such as the one shown in Figure 4.

Having students complete the experiment in a 50-minute or 75-minute class period is very challenging, especially since this is the first time that they have been exposed to protoboards. To streamline the in-class experiment, students must complete a pre-lab before class where they design the state machine logic and circuit. Students are also asked to view a seven-minute instructional video that gives an introduction to protoboards and to the experimental platform. These platforms are partially built and pre-wired for the students. The platform is given to students in the form shown in Figure 1: the protoboard with the battery pack mounted, chips inserted, dip switches inserted with appropriate pull down resistors, push buttons (for toggle and clear) with antibounce filters. The ground and high voltage are pre-wired as well as the enable, the clock and the reset lines. In this way, the student can see the implementation aspects, such as debouncing, but need concentrate only on the connection of wires that implement the state machine transition logic.

In addition to the experimental platform, the components of the Finite State Machine Module are:

Fundamental Concepts tutorial – containing a summary of state transition diagrams and state tables, decoders, registers, and state machine circuit.

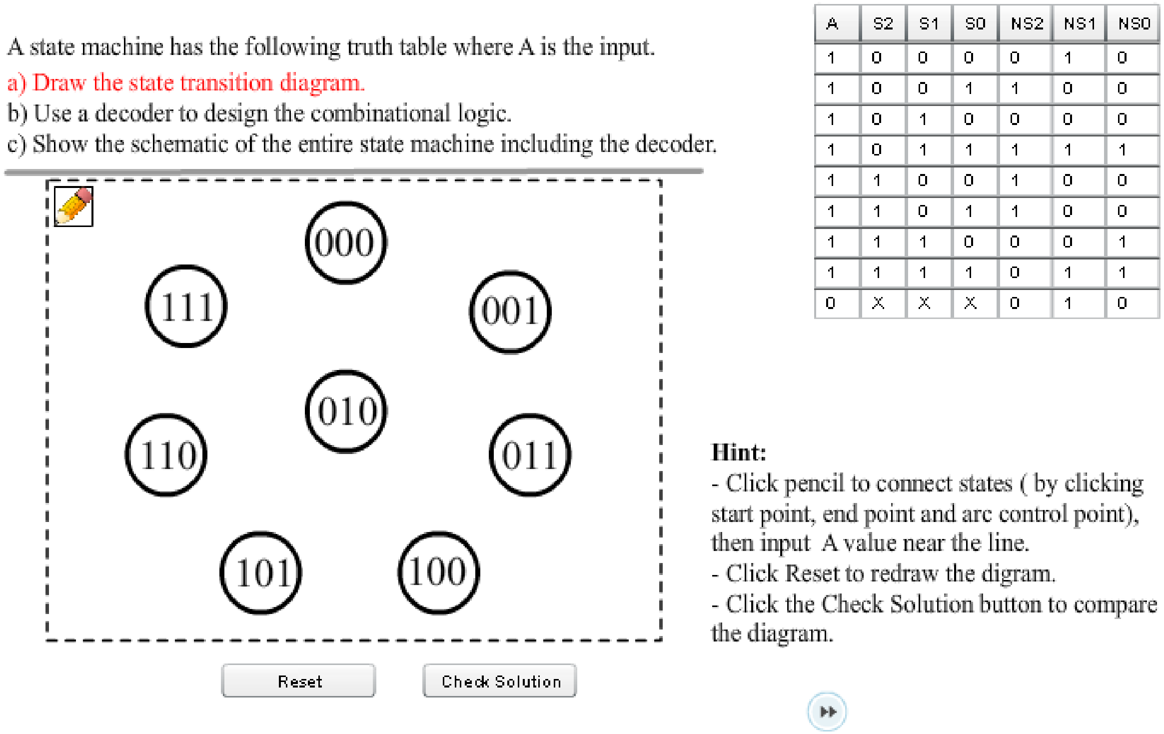
Pre-lab and Lab Instructions – the prelab is to be completed prior to class and the remainder of the lab is to be completed in class

Instructional Video – introduction to protoboards and how to wire the example circuit; video is available for viewing online or for downloading to an IPOD.

Virtual Experiment – a Flash demo is available that is built upon the image of the partially wired protoboard shown in Figure 1. Students can use their mouse to complete the wiring for the example state machine. They can then test the circuit's operation by clicking on the input dip switch (to toggle between an input of 0 and 1), and the circuit's clock button. The LEDs "light up" in the image according to the example state machine logic.

Quiz Questions – questions representative of those found in tests on this material modified to be completed in an on-line manner using Flash. A sample quiz question is shown in Figure 5. Note that there are several parts to this problem, and the figure only shows the first part.

Instructor Resources – parts list, schematic of circuit, best practices for implementation, FAQ for common student mistakes


A state machine has the following truth table where A is the input.

a) Draw the state transition diagram.

b) Use a decoder to design the combinational logic.

c) Show the schematic of the entire state machine including the decoder.

A	S2	S1	S0	NS2	NS1	NS0
1	0	0	0	0	1	0
1	0	0	1	1	0	0
1	0	1	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1
1	1	1	1	0	1	1
0	X	X	X	0	1	0

Hint:
- Click pencil to connect states (by clicking start point, end point and arc control point), then input A value near the line.
- Click Reset to redraw the digram.
- Click the Check Solution button to compare the diagram.

Figure 5: Online quiz question, written in Flash.

The video takes students through the pin diagram shown in Figure 3 and gives them the introductory steps needed to implement the sample state machine on the partially wired protoboard shown in Figure 1. Students may try to wire the circuit prior to class by using the virtual experiment built using Flash.

A 75-minute lecture (typically, a Tuesday-Thursday class) is long enough to run the experiment for the state machine shown in Figure 4. Students start finishing at 30 minutes with the majority finishing around 50-60 minutes. It takes some students the entire class period to finish the experiment. Alternate forms of the state machine are available that are a little simpler and would require less time, hence would be suitable for a 50-minute lecture. The difference is that the machine shown in Figure 4 uses a 3-input OR gate while the example circuit and the alternative machines use only 2-input OR gates. This added complication of the 3-input OR gate is harder for some students than others to comprehend; hence the large variation in completion time.

Logistics

Students are given the website for the State Machine module and asked to follow these steps:

1. Read the Fundamental Concepts document
2. Download and print the Lab Instructions (includes the prelab)
3. Complete the state machine design in the pre-lab
4. View the video (the description of pin diagrams is useful for completing the pin diagram required in the pre-lab)
5. Practice the wiring procedure on the virtual experiment and test its operation
6. Bring the lab instructions to class and complete the lab in class (wire the example circuit and their designed circuit)
7. Do the online quizzes for the material

At the beginning of class, students are partnered with their neighbor (2-person groups), and each group received one board and 8-10 extra wires. The introductory video is shown for those who need it. They are asked to complete the lab in-class and turn in one lab write-up per group.

Teaching Assistants are extremely helpful for smooth implementation of this experiment. First, they are requested to run through the lab before class and to test the boards. They also bring the boards to class. It is recommended that one TA be available in the class for every 10 groups. (The course instructor can substitute for one TA.) Students must demonstrate their circuits, both the example circuit and their designed circuit, to the TAs. The demonstration, consists of running through a set sequence of inputs and recording the resulting outputs. The TAs answer questions and sign off on the demonstrations.

Results and Assessment:

This experiment has been used in 11 classes of Introduction to Computer Engineering over four semesters,. This is a required course for both Computer Engineering and Electrical Engineering majors. .The course is a technical elective for Industrial and Systems Engineering majors, so the makeup of any particular class tends to be a mix of these majors.

By integrating the State Machine Module in this course, students become active learners. Since the experiment allows for students to work in pairs in a problem-solving format, the value-added benefit of using the module is even greater. The assessment of using the module consists of two primary strategies: reviewing test performance on those questions that correspond to the material covered by the experiment and conducting student surveys about their interest and understanding of the material. The assessment includes four control and four experimental classes. The full assessment analysis will compare students from the experimental group (those in classes using the hands-on experiment) and from the control group (those in classes using the traditional instruction) using selected demographic variables in order to consider other possible explanations for differences in tests performance. For each of the comparison classes, the same instructor taught both the experimental and control courses.

Preliminary survey data shows that students in the experimental classes have a better understanding of the state machine material than in the traditional class. For six classes (three pairs of experimental and control classes and each pair taught by the same instructor), higher percentages of students in each of the control classes report that they did not understand the state machine material as well as other material in the class. Between 60% and 70% of students in each control class reported this compared to 31.7%, 40% and 47% in the classes using the modules. In one of the pairs, 41% of the students in the experimental class responded that they had a better understanding of the state machine material than other material compared to 19% in the control class. Additional analysis will include comparisons of test performance on the specified questions about state machines. Taken together, both assessment tools, comparative analysis of student performance and student attitudinal data, will address the effectiveness of the hands-on experiments to enhance student learning.

Acknowledgements:

The authors thank the following professors who have used the experiments in their classes and made many suggestions on its improvements: D. Schimmel, A. Chatterjee, S. Yalamanchili, D. Blough, and A. Howard. The authors also thank Rachel Laskin for her help in reviewing the documentation for the lab. This projects was funded under an NSF CCLI grant.

References:

(Dua, 2005), Dua, R., Seiffertt, J.E., Blaha, B., Gupta, K., Satagopan, V., Stanley, J.R., Beetner, D., and Wunsch, D.C., 2005, “Hands-On Projects and Exercised to Strengthen Understanding of Basic Computer Engineering Concepts,” Proceedings of the American Society of Engineering Education Annual Conference & Exposition, Portland.

(Durfee, 2005), Durfee, W., Li, P., and Waletzko, D., 2005, “At-Home System and Controls Laboratories,” Proceedings of the American Society of Engineering Education Annual Conference & Exposition, Portland.

(Ferri, 2008), B. Ferri, J. Auerbach, J. Jackson, J. Michaels, D. Williams, “A Program For Distributed Laboratories In The ECE Curriculum,” 2008 ASEE Frontiers in Education Conference, June 22-25, Pittsburgh.

(Ferri, 2009), Ferri, B.H, Ahmed, S. Michaels, J., Dean, E., Garvet, C., Shearman, S., “Signal Processig Experiments Wit LEGO MINSTORMS NXT Kit for Use in Signals and Systems Courses,” *Proceedings of the American Control Conference*, St. Louis, pp. 3787-3792.

(Hendricks, 2005), Hendricks, R.W., Lai K.M., and Webb, J.B., 2005,”Lab-in-a-Box: Experiments in Electronic Circuits That Support Introductory Courses for Electrical and Computer Engineers,” Proceedings of the American Society of Engineering Education Annual Conference & Exposition, Portland.

(Java, 2010), “Finite State Machine Simulator,” based on an applet by Kyril Faenov http://www.cs.virginia.edu/~akn8c/t/ accessed 3/1/10.

(Litwhiler, 2005), Litwhiler, D.H., Lovell, T.D., 2005, “Acoustic Measurements Using Common Computer Accessories: Do Try This at Home,” Proceedings of the American Society of Engineering Education Annual Conference & Exposition, Portland.

(Mathworks, 2010), “Overview of Mealy and Moore Machines,” http://www.mathworks.com/access/helpdesk/help/toolbox/stateflow/ug/bqtktf3-1.html, The Mathworks, accessed 3/1/10.