Virtual Experiments for Digital Controller Design Projects

Prawat Nagvajara
Department of Electrical and Computer Engineering
Drexel University

Abstract

We are developing a set of software applications that simulate and animate physical systems such as traffic at an intersection, and monorail and elevator systems. We call the software applications *"virtual experiments,"* and use them to teach digital controller design. These software applications run on a PC or a Macintosh to provide real-time simulations and animation which interact with an external controller via the computer's IO port. The students design different problems in which the controller receives information about the state of the system and sends the control signals to the virtual experiment software application. The students can solve these control problems using software solutions, e.g., via a microcontroller, and hardware/firmware solutions via Field Programmable Gate Array (FPGA). The courses that can use this teaching tool include design with microcontroller and digital systems (FPGA) design.

I. Introduction

Embedded systems design and top-down digital design (using hardware description language and design automation tools) are gaining popularity (and becoming necessary) in the electrical and computer engineering curriculum due to the demands in current technology. To that end, challenging hands-on design projects, which now accompany the courses, prove very effective teaching tools. Applications of digital systems in areas such as signal processing, telecommunications and high-performance computers represent plausible design problems, however, the control problems in which physical systems interact with digital systems to accomplish certain tasks, offer in our opinion more exciting hands-on design projects than other applications. With emphasis on digital design, students can produce a viable solution to simple control problems with minimal knowledge in control theory by using their intuition.

To reduce the overhead in constructing physical systems for our digital controller design projects, we adopt the use of computer simulations to create *"virtual experiments"* to replace the actual physical systems. The advantage is that we only require the development cost. Further, dissemination of this teaching aid can be done via the Internet. There do exist projects such as mobile robot, aerial robot and underwater robot in which simulation and animation cannot match

the excitement of the real world, nevertheless, simulation to aid the design could prove valuable. The control problems we are developing into virtual experiment software include the traffic light, the monorail and the elevator systems.

We have developed virtual experiments for teaching digital systems design, in which we categorize the implementations of the controllers in terms of both a software solution and a hardware/firmware solution. In the case of the software solution, the students use a micrcontroller and develop the software that interfaces with the virtual experiment software, as part of a microcontroller course. In the case of the hardware/firmware solution, the students use Field Programmable Gate Arrays (FPGAs) (and memory) to implement a digital controller. This type of activity can be integrated into top-down digital (FPGA) design courses. By doing these projects, the students are able to practice and understand key concepts in embedded systems design, namely, object-oriented system modeling, concurrency, top-down digital design and interfaces. The students gain hands-on experience with high-level programming language and design automation technology. Additionally, they gain experience with solving engineering problems including policy (procedural) control problems, as well as some dynamic control problems. We assume that the students have only studied elementary dynamics (from physics and system courses) and are capable of further research on a more complex dynamic model, e.g., a moving body and DC motor.

II. Virtual Experiments

Below we describe a set of *"virtual experiments,"* software applications that simulate and animate physical systems such as traffic at an intersection, and monorail and elevator systems. These software applications run on a PC or a Macintosh to provide a real-time simulation of physical systems and an interaction with the external controller via the computer's IO port.

A. Traffic light Control Experiment

We have completed the traffic light simulation and animation software and have been using the application in teaching design in a microcontroller course and an advanced top-down digital (FPGA) design course. This application was developed in Java. The software provides animation of an intersection where cartoon-like cars travel in north/south and east/west directions and obey the traffic light. The students design an external controller to control the traffic lights inside the animation. The external controller receives two signals from the computer running the virtual experiment software, (1) indicating the presence of cars waiting at the red light, and (2) acknowledging the changing of lights upon the receipt of a request from the controller. The external controller can send signals to the computer running the animation to change the lights at the intersection. The interface between the virtual experiment application running on the computer (Macintosh) and the external controller is via an interface unit called ADB I/O

(Beehive Technology Inc.) [1], in which a driver for Java is available. The students also learn how to implement an asynchronous (handshake) protocol between the Macintosh and the external controller.

As an additional visual effect, we have an actual set of traffic lights (purchased from a vendor) and relays accompanying the virtual traffic. We display the animation onto a large screen using a video projector. This provides visual effects of the virtual experiments.

We use this traffic light experiment in two courses: *Design with Microcontrollers* and *Design for Synthesis and Testability, and Performance Modeling* (senior/graduate). In the *Design with Microcontrollers* course (pre-junior), the students code a microcontroller, namely the *Handyboard* and *Interactive C* (developed at MIT Media Lab) [2], to implement the controller. They learn to design using multi-tasking, which includes a finite state machine, timers, handshake protocol, and polling tasks (polling the values of signals from the virtual experiment). The students break down the systems into tasks, and learn about client-server models and inter-process synchronization.

In the *Design for Synthesis and Testability, and Performance Modeling* (senior/graduate) course, the students design the controller using VHDL [3], simulated and synthesized to an FPGA *(Xilinx XC4000E).* The system includes an internal clock, counter for clock divider, finite state machine and handshake protocol unit.

B. Monorail and Elevator Control Experiments:

With the success of the traffic light simulator project for microcontrollers and FPGA design, we are now developing monorail and elevator experiments. In the monorail system the virtual experiment software provides a simulation of a dynamic model, describing the voltage driving force and damping (break) to a dc motor and moving train. The graphic animation displays the moving train according to the dynamic model. The control signals consist of the voltage to the motor and the damping force (break). The virtual experiment provides the position of the train to the external controller. Similarly, in the elevator system we have a dynamic model of the moving elevator car with countered weight and dc motor.

The object of the monorail problem is to provide service to passengers (transporting) between a sequence of terminals- A, B, C, D and E. Actual systems of this type already exist at some airports. The distances and the number of passengers vary from terminal to terminal. The controller will control the opening and closing of the door when all passengers embark the train indicated by a signal available to the controller. In the elevator control problem, we have a two-car elevator system, with animation showing the cars, the floors and the people (users).

We have divided the software design of the monorail and elevator into separate modules: (1) the dynamics model, (2) the characters (passengers), (3) the simulation and animation, and (4) the interface to the external controller. These modules are executed in an environment developed in Tcl/Tk shell (Scriptics Inc.) [4]. The modules are developed in C++ and, the animation is implemented using the built-in routines within Tcl, in order to achieve a smooth animation and update of display data. (Note: That we do not use an interpretive language like Java.) With regard to the portability of these applications, we do not expect difficulty in compiling the source codes on most platforms.

III. Conclusions

The student projects on controllers for monorail and elevator systems that optimize service time, represent challenging tasks, whether the implementations is based on a software solution or an even more complex hardware/firmware solution. We have successfully completed the less complicated project of traffic light control and are tackling the more complicated monorail and most complicated elevator control problems. We plan to have these virtual experiment software applications available on our FTP site in the near future.

Bibliography

1 Beehive Technologies Inc., URL http://www.bzzzzzz.com

2 F.G. Martin, *The Handyboard Board Technical Reference*, URL http://lcs.www.media.mit.edu/groups/el/projects/handy-board/

3 J. Bhasker, *VHDL Primer*, Third Edition, Upper Saddle River, NJ: Prentice Hall, 1999.

4 Scriptics Inc., URL http://www.scriptics.com and http://www.tcltk.com

PRAWAT NAGVAJARA

Prawat Nagvajara is an Associate Professor of Electrical and Computer Engineering at Drexel University. He is an educator in Computer Engineering including Top-down Digital Design, Hardware Description Languages and Design Automation, Embedded systems, and Computing System Security. He received a B.S. Degree and M.S. Degree from Northeastern University, and a Ph.D. from Boston University in 1989, all in Electrical and Computer Engineering. His research interests include Application Specific Integrated Circuit Design, and Design for Testability.