

# Using FPGAs to Simulate and Implement Digital Design Systems in the Classroom

Tyson S. Hall<sup>1</sup> and James O. Hamblen<sup>2</sup>

**Abstract**—Field-Programmable Gate Arrays (FPGAs) have gained widespread popularity in digital design laboratories. Their flexibility and relatively low cost make them ideal pedagogical resources. When FPGAs are used as the primary platform in a digital design laboratory, increased flexibility and topical integration is provided to the instructor. Topics that have traditionally been taught in separate, independent laboratory assignments can be bridged together to form a cohesive series of laboratory projects based on a single problem scenario. This paper will present a series of laboratory exercises that center around a model train track with multiple tracks, trains, sensors, and switches.

*Keywords:* Field-programmable gate array, FPGA, digital design laboratory.

## I. INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) are becoming increasingly commonplace in digital design laboratories at all course levels. FPGAs have been shown to afford a number of new opportunities for classroom learning. FPGA-based robotics have made intriguing laboratory experiments in entry-level digital design courses up through senior-level capstone design courses [1], [2], [3]. FPGAs have also been successfully used in the classroom to study system-on-a-programmable-chip (SOPC) design, hardware/software co-design, computer architecture, and signal processing hardware implementations [4], [2], [5].

Entry-level digital design courses can use FPGAs in a laboratory setting to synthesize basic combinational logic circuits, counters, finite state machines, simple computer datapaths, memory interfaces, and more. The flexibility and ease of use of FPGAs provides an opportunity for students to work on more meaningful projects with tens of thousands of gates while still learning the fundamentals of digital design [6], [7]. If the instructor provides interface code to sensors, motors and I/O devices, students can design simple logic circuits that control autonomous robots, VGA simulations, and even HO model trains. By providing more realistic problem scenarios, students are able to both learn the digital design concepts and gain an appreciation for large-scale digital design problems.

This paper will present a series of laboratory exercises that center around a model train track with multiple tracks, trains, sensors, and switches. Using this same model train setup, laboratory projects can be written that emphasize hardware description languages (VHDL or Verilog), finite state machine development, and simple computer datapaths with the accompanying assembly language programming.

An FPGA development board from Altera Corporation is used to implement both a VGA simulation engine for testing students' projects and an interface to a physical HO model train track with Digital Command Control (DCC) capable trains. Thus, a single FPGA development board can be used to both simulate the problem scenario and then implement a control interface to a physical implementation of the same situation.

## II. FPGA HARDWARE

FPGAs are ideal devices for student laboratory projects. Since they can be easily reconfigured, FPGAs can support a wide variety of projects and can be reused almost indefinitely. The computer aided design (CAD) software needed for FPGA development is available free for school laboratories and even for student-owned PCs. The two major

---

<sup>1</sup> Southern Adventist University, Collegedale, TN 37315-0370, tyson@southern.edu

<sup>2</sup> Georgia Institute of Technology, Atlanta, GA 30332-0250, hamblen@ece.gatech.edu

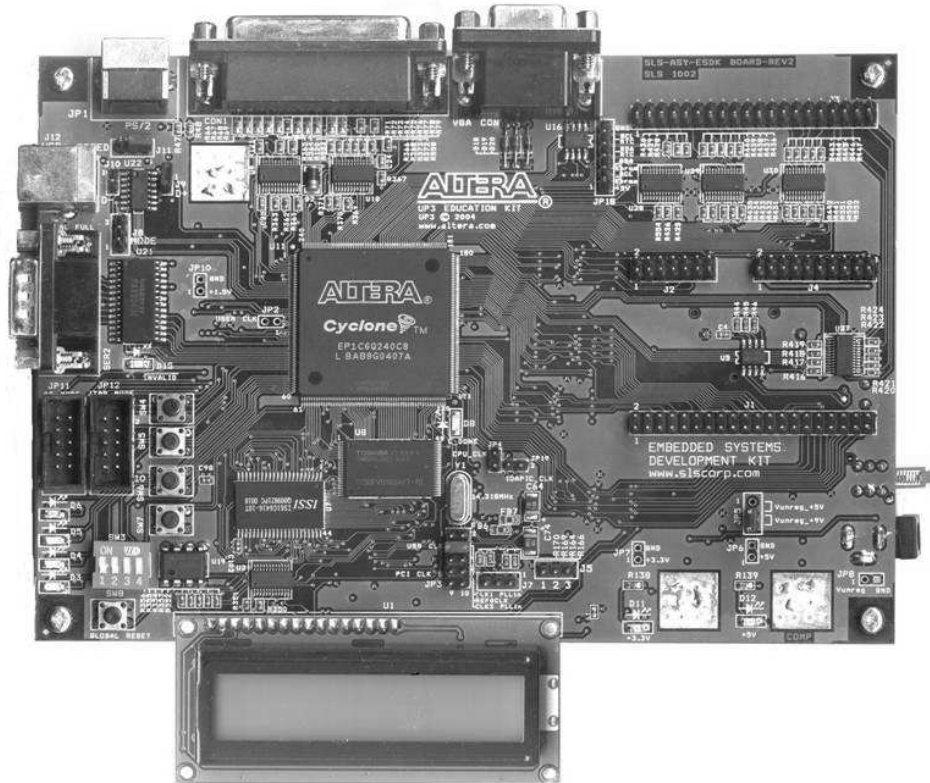


Fig. 1. A current generation FPGA development board contains a large FPGA, external memory, and a wide variety of I/O devices. An Altera UP3 board is shown.

FPGA vendors, Altera and Xilinx, distribute their software freely over the web. For school-owned laboratory PCs, instructors can get a free license for the commercial version of the software through Altera's and Xilinx's University Programs [8], [9]. Some third-party add-on tools and intellectual property (IP) cores used only in more advanced designs are more difficult for schools to obtain, but they are not needed for introductory courses.

The basic FPGA CAD tools support design entry using schematic capture, entry with hardware description language (HDL) based models using VHDL or Verilog, logic synthesis, and logic simulation. Logic simulators support zero delay models for functional simulation, and they also have full timing simulations based on actual delays from timing models of the FPGA being used. The CAD tools also provide an easy interface to program the FPGA. Schematic capture is rarely used since it is too slow and cumbersome for large designs. Design entry using HDLs greatly increases productivity and is commonplace in industry. VHDL has a Pascal or ADA style syntax, and Verilog syntax is similar to C. VHDL is more widely used in schools since it was available earlier in the public domain. In industry, VHDL and Verilog currently have an almost equal share of new designs.

The new generation high pin count chip packages used in current FPGAs cannot be interfaced easily to standard student laboratory protoboards. Low-cost FPGA development boards are typically used to avoid this problem. The Altera UP3 board is one such FPGA development board designed for students, and a picture of it is shown in Fig. 1 [8]. This board contains a large FPGA with a gate equivalency of several hundred thousand gates, volatile and non-volatile memory, and a wide assortment of I/O hardware. Common I/O features include LEDs, switches, LCD, PS/2, VGA, audio, and Ethernet. A parallel or USB interface is provided on the board to program the FPGA from a development PC. Similar boards are also available from Xilinx [9], [10]. With the educational discounts provided by FPGA vendors, the price can be as low as a contemporary textbook. This even makes it possible to consider laboratory options where students buy their own board. The lowest cost FPGA boards have fewer I/O

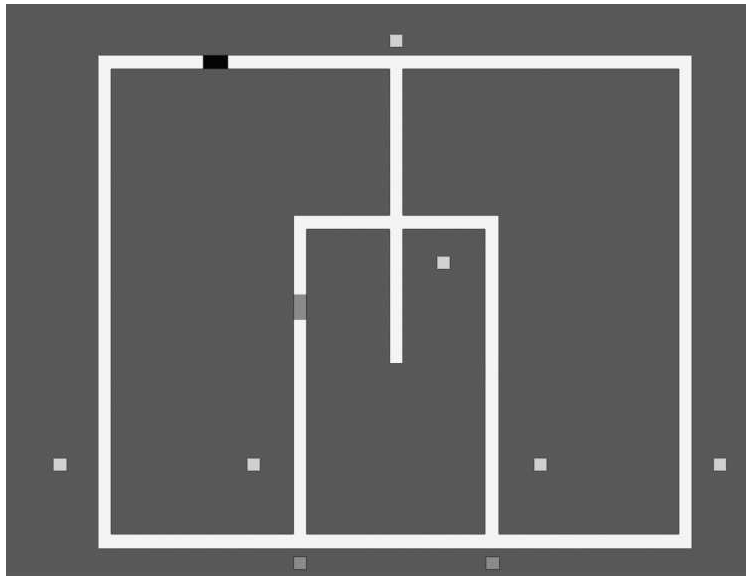


Fig. 2. This is a sample of the graphical view of the tracks, trains, sensors, and switches that is output to a VGA monitor by the train simulator. Students write state machine controllers, which guide the trains around the tracks in a defined pattern while avoiding collisions and derailings.

options and a smaller FPGA, but even these boards typically support around one hundred connections from external devices to FPGA pins using PCB header connectors. A standard ribbon cable connector can be used to connect the FPGA to external hardware on a custom PCB or to a standard laboratory protoboard. When interfacing to external devices, be aware that some of the newest FPGA devices use 3.3V digital logic levels instead of the older 5V logic levels.

### III. TRAIN SIMULATION

Most digital logic laboratory courses start by implementing simple combinational circuits and then study basic sequential digital circuit elements such as latches and D flip-flops. LEDs and switches connected to FPGA logic on an FPGA development board work well for these very simple early projects. The next topic in digital laboratory courses is typically state machines and state machine design techniques. For most students, this is the most difficult portion of any digital logic course. Using state machine examples based on concrete physical examples that students already understand helps in the learning process. In entry-level digital design laboratories, a simple train system can be used as the state machine design example. Students intrinsically understand trains and how they must operate safely on tracks and switches. They also enjoy video games, and most are intrigued with model train systems. This laboratory project utilizes all of these ideas.

Figure 2 shows a video image of the train simulation system setup. Two trains are present on the tracks. The track layout has two circular sections of track with the smaller circular track inside of the larger one. A small track spur off of the top of the outer track also crosses the inner track. Three track switches are located at track intersections and five IR sensors indicate the presence of a train on each track segment between switches.

Students design a state machine that is used as a safety controller for the train system. It allows both trains to move in and out of intersections without mishap. The state machine's outputs control the power to the tracks, the direction of the trains, and the position of the track switches. By reading the sensor inputs, the state machine can determine the track segment where each train is currently located. For safety, only one train at a time can be present in a track segment. The state machine does not control the relative speed of the two trains.

Students are provided with a detailed verbal description of how the train system should operate and are then required to design their own state machine for the train system. The controller must successfully guide the trains in the defined pattern while avoiding any possibility of collisions. A state machine design that uses a simple pattern to operate the train system is provided as an example. Both a state diagram and a VHDL implementation of the

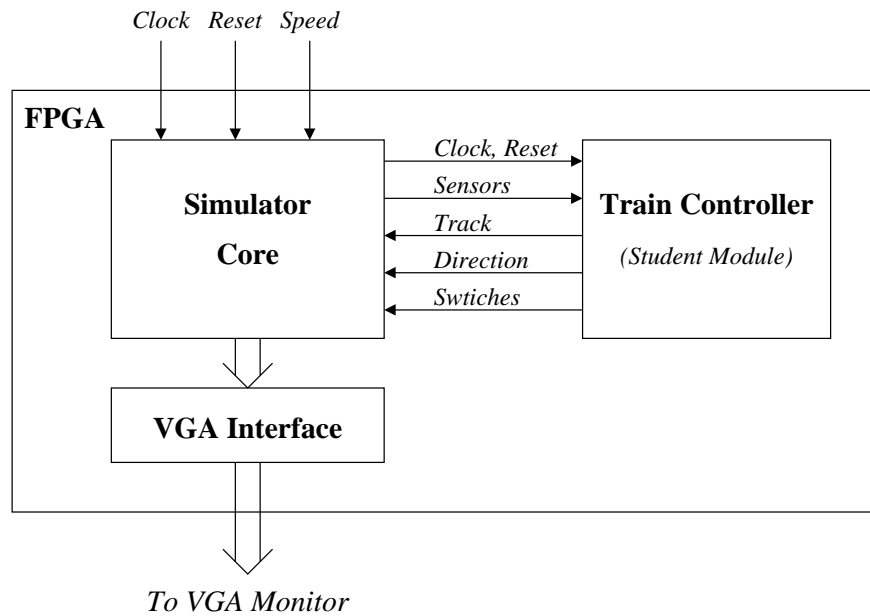


Fig. 3. This is a block diagram of the VHDL modules involved in the VGA train simulation. The simulation core and VGA interface are provided for the students in a top-level project file to allow them to focus on their controller module.

state machine are included in the example. The train system is complex enough that a new design problem can be generated each semester. For example, one semester trains could move clockwise and another counter-clockwise. The path for the trains to follow and priority of each train can also be varied. The small track spur and its associated switch provide additional design problem variations.

To simulate students' new state machine design, a video virtual train simulator is encapsulated within an IP core that is given to the students. A block diagram of the simulator system is shown in Fig. 3. The simulator IP core is connected to their state machine and both fit on a single FPGA. The IP core generates inputs, monitors outputs, and produces a VGA display output on the FPGA development board. The video output from the IP core seen in Fig. 2 is a graphical representation of the state of the train system. The trains move around the track image and settings of sensors and switches are indicated with different colors. Running the simulation with one train running fast and the other running slow makes design problems appear faster in the simulation. If the IP core detects a safety violation (such as two trains on the same track segment or derailling by going backwards through an open track switch), it stops and flashes the video display at the point the problem occurred.

Students are required to successfully demonstrate their controller design on the virtual train simulations to a laboratory TAs before they are allowed to run their state machine design on the physical model train system that is described in the next section.

#### IV. TRAIN IMPLEMENTATION

Traditionally, model train tracks that allow individual control of multiple trains have required isolated track segments, multiple power supplies, and a relay network to switch the power supplies between the multiple track segments. This track scheme can be complex to build and maintain in an academic setting, and the relay network is susceptible to damage if students' controller modules are not designed correctly. Digitally controllable trains, however, are now readily available making model train tracks an attractive option for digital design laboratory experiments.

The train setup that is used in the train simulator described in the previous section has been built using a digitally controllable HO model train system. The track, sensors, and switches are located in similar positions to the simulator, and the power supply signals (speed and direction) are converted to digital commands that are sent to each of the trains. A second IP core is provided to the students with the same interface as the simulator to ease the transition.

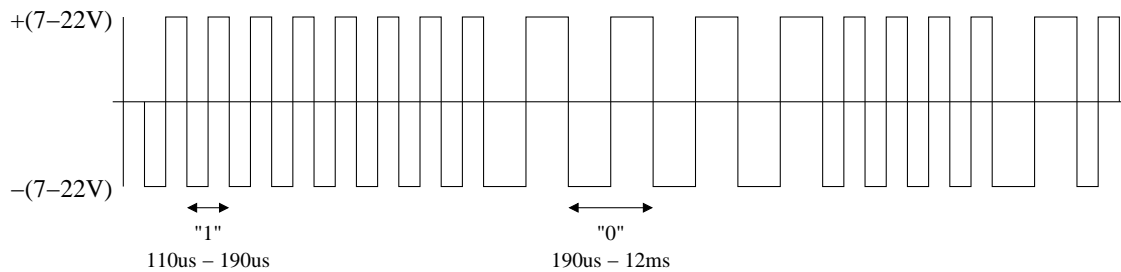


Fig. 4. This is a sample DCC waveform. The DCC serial bit stream is transmitted on the train tracks in a bipolar manner as shown. The time between zero crossings determines the value of each bit.

This IP core implements the interface between students' controller and the physical sensors, switches, and train tracks.

#### A. DIGITAL COMMAND CONTROL

The National Model Railroad Association (NMRA) has approved a Digital Command Control (DCC) standard that defines a completely digital model train system. On a DCC system, trains are individually addressable. Train speed, direction, and special features are controlled via a bipolar digital bit stream that is transmitted on the train tracks (see Fig. 4). A DCC decoder is located on each train that interprets the serial commands and initiates the desired action (i.e., change in speed, direction, or feature status). Trains are typically powered by a full-wave rectification of the bipolar data signal that is present on the track.

An FPGA can send the DCC data streams to the train track with an H-bridge circuit, and integrated H-bridge modules are available that can minimize the number of discrete components used. The National Semiconductor LMD18200 integrated H-bridge module was used for this implementation. The LMD18200 supports TTL and CMOS compatible inputs allowing the Altera FPGA board to be connected directly to this chip.

#### B. TRAIN TRACK SETUP

The total cost for the train, track, train programmer/controller, and remote switches was \$300. The Atlas True-Track nickel silver train track with roadbed was used for this implementation along with Bachmann DCC-enabled train locomotives. In addition to trains and track, three Atlas electrically controllable switches were needed, and a Bachmann command controller was purchased to program the trains' addresses and test the setup.

Figure 5 illustrates the layout for the train track setup. This track configuration requires a relatively level surface that measures five by six feet in dimension. Rerailers are placed on the tracks close to the starting position of the two trains. This aids in quickly resetting the system to the proper original state when several student groups are using it during a single laboratory period.

#### C. SENSORS

Sharp GP2L26 photointerrupter sensors are used to detect when a train passes each sensor point. A picture of this sensor is shown in Fig. 6. These sensors are very small (3mm x 4mm) and fit between the rails on the track. Wires are run down through the roadbed to a central protoboard where the discrete components needed for interfacing to this sensor are connected. The model trains used in this example had primary black underbodies, and the photo sensors can not always detect the trains passing over them. To increase the visibility of the trains to the photo sensors, pieces of white paper were taped to the bottom of both trains. This dramatically increased the accuracy of the sensors.

#### D. TRAIN CONTROL MODULE

A block diagram of the complete model train control system is shown in Fig. 7. The complete control logic for this setup resides on the FPGA. Also, it was designed to have an identical interface to the students' module as

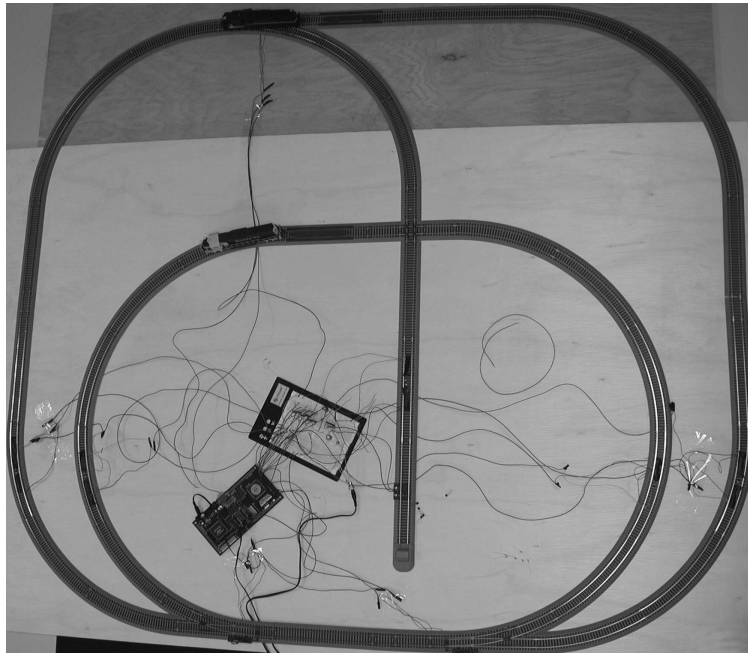


Fig. 5. This is a picture of the train track layout.



Fig. 6. The Sharp GP2L26 photointerrupter sensors are 3mm x 4mm, and they are located between the rails of the track at five points around the track layout. Each sensor can detect when a train passes over its track position.

the virtual train simulation system (see Fig. 3). The train control module is written in VHDL, and it includes the necessary logic to output the DCC data stream, control the switches, and input the sensor values.

The DCC data stream is generated by combining the *Speed* input and the *Direction* from the student's module. The appropriate DCC command is created from these signals and then registered. The registered command is shifted out in a serial stream. The DCC standard only provides for one-way communication, and thus, no transmission guarantee can be made. Therefore, a given command is repeatedly shifted out until another command is received to ensure transmission of each command. Continuous transmission also insures a consistent power level on the tracks.

The Atlas remote switches are activated by providing a short pulse (<1 sec.) of +12V power to one of the two control pins—one control pin moves the switch to connect the inside track and the other control pin moves the switch to connect the outside track. The *Switch* signals generated by the students' controller module are monitored for rising and falling edges. When a new value is detected, the train control module sends a 100 ms pulse to the appropriate switch control signal.

## V. CONCLUSION

FPGAs have provided a wide variety of new opportunities for academia. FPGAs are used to both simulate and then control an actual model train system. The large scope and variety of laboratory projects that can be created using this infrastructure make it an attractive option for entry-level digital design laboratories. In an earlier incarnation of

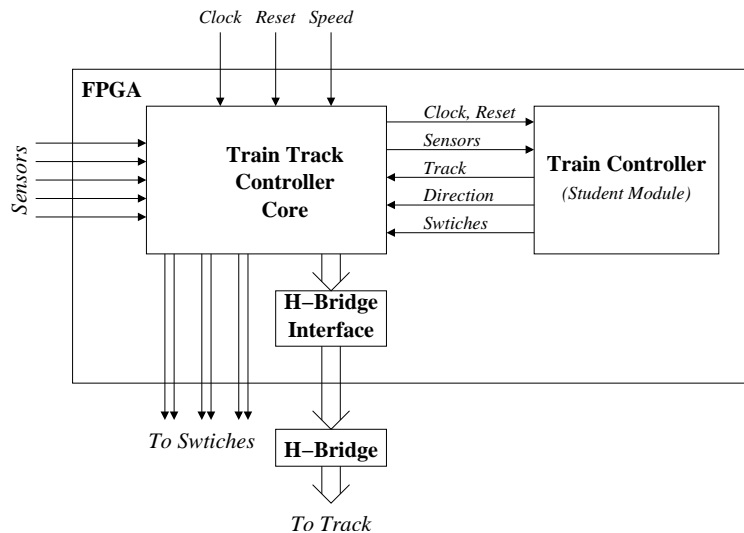


Fig. 7. This is a block diagram of the VHDL modules involved in the model train track controller. The train track controller core and H-Bridge interface are provided for the students in a top-level project file to allow them to focus on their controller module. The interface to the students' module is identical to the one in the VGA train simulation (see Fig. 3) allowing the same student module to be used for both the simulation and the physical train track implementation.

the train design project before FPGAs and CAD tools, students used the model train system without simulations. However, the model trains wrecked so often that they rarely survived a semester. A modern DCC-capable train system has allowed the physical system to be used again. Seeing their state machine control real hardware adds to students' experience and to their overall sense of accomplishment. Using it in conjunction with the VGA train simulation protects the physical train setup from injury, and it also reinforces the need for simulation and testing.

## VI. ACKNOWLEDGEMENTS

Many students and teaching assistants have contributed to this work during the past few years. In particular, we would like to acknowledge Nick Clark, Ivan Delgado, Zachary Folkerts, James Johnston, Andrew Mayers, Michael Neubrander, and Chris Walters for their help in building and testing the physical model train setup.

## References

- [1] K. Newman, J. O. Hamblen, and T. S. Hall, "An introductory digital design course using a low-cost autonomous robot," *IEEE Transactions on Education*, vol. 45, no. 3, pp. 289–296, Aug. 2002.
- [2] J. O. Hamblen, "Rapid prototyping using field-programmable logic devices," *IEEE Micro*, vol. 20, no. 3, pp. 29–37, May/June 2000.
- [3] J. O. Hamblen and T. S. Hall, "Engaging undergraduate students with robotic design projects," in *Proceedings of the Second IEEE International Workshop of Electronic Design, Test and Applications*, Jan. 2004.
- [4] T. S. Hall and J. O. Hamblen, "System-on-a-programmable-chip development platforms in the classroom," *IEEE Transactions on Education*, vol. 47, no. 4, pp. 502–507, Nov. 2004.
- [5] T. S. Hall and D. V. Anderson, "A framework for teaching real-time digital signal processing with field-programmable gate arrays," *IEEE Transactions on Education*, vol. 48, no. 3, Aug. 2005.
- [6] M. A. Soderstrand, "Role of FPGAs in undergraduate project courses," in *1997 IEEE International Conference on Microelectronic Systems Education Proceedings*, April 1997, pp. 109–110.
- [7] M. S. Nixon, "On a programmable approach to introducing digital design," *IEEE Transactions on Education*, vol. 40, no. 3, pp. 195–206, Aug. 1997.
- [8] Altera Corporation, <http://www.altera.com/>, *Embedded Processor Website*, 2005.

- [9] Xilinx Corporation, <http://www.xilinx.com/>, *Embedded Processor Website*, 2005.
- [10] Digilent Corporation, <http://www.digilentinc.com/>, *Home page*, 2005.

**Tyson S. Hall**

Tyson Hall received the PhD, MSECE, and BSCMPE degrees in electrical and computer engineering from the Georgia Institute of Technology in 2004, 2001, and 1999. He is currently an Assistant Professor in the School of Computing at Southern Adventist University in Collegedale, Tennessee. Dr. Hall's research interests include rapid prototyping of mixed-signal systems, cooperative analog/digital signal processing, reconfigurable computing, and embedded systems education.

**James O. Hamblen**

James Hamblen is a Professor in Electrical and Computer Engineering at the Georgia Institute of Technology. His current research interests include rapid prototyping, high-speed parallel and VLSI computer architectures, computer-aided design, and reconfigurable computing. Dr. Hamblen received a PhD in Electrical Engineering from Georgia Tech, an MSEE from Purdue University, and a BEE from Georgia Tech. Prior to earning a PhD, he worked as a Systems Analyst for Texas Instruments in Austin, Texas and as a Senior Engineer for Martin Marietta in Denver, Colorado.