# An Introductory Digital Design Course Using a Low-Cost Autonomous Robot

Kimberly E. Newman, *Member, IEEE*, James O. Hamblen, *Senior Member, IEEE*, and Tyson S. Hall, *Student Member, IEEE*

***Abstract*—This paper describes a new digital design laboratory developed for undergraduate students in this electrical and computer engineering curriculum. A top-down rapid prototyping approach with commercial computer-aided design tools and field-programmable logic devices (FPLDs) is used for laboratory projects. Students begin with traditional transistor–transistor logic-based projects containing a few gates and progress to designing a simple 16-bit computer, using very high-speed integrated circuits hardware description language (VHDL) synthesis tools and an FPLD. To help motivate students, the simple computer design is programmed to control a small autonomous robot with two servo drive motors and several sensors. The laboratory concludes with a team-based design project using the robot.**

***Index Terms*—Autonomous robot, digital design, field-programmable logic device (FPLD), sequential logic, very high-speed integrated circuits hardware description language (VHDL) synthesis.**

## I. Introduction

THE NATURE and background of undergraduate students entering electrical and computer engineering (ECE) programs have changed significantly in recent decades. Traditional pedagogical methods of teaching theory before looking at applications do not adequately address the needs of today's students [1], [2]. Without previous, relevant experience, students are not motivated to learn the material in the fundamental engineering courses that traditionally fill the students' first several years of course work [1]. In an effort to appeal to this new generation of engineering students, promote higher retention rates, and increase motivation in undergraduate students, the faculty at the Georgia Institute of Technology (Georgia Tech) have redesigned the computer engineering curricula to follow a more top-down approach [2].

Traditionally, transistor–transistor logic (TTL) protoboard-based projects have formed the backbone of digital design laboratories. Because of the time-intensive nature of designing with this technology, projects in an introductory laboratory are limited to a handful of gates. This limitation is unacceptable when trying to introduce students to more interesting real-world applications. Alternatively, field-programmable logic devices (FPLDs), which include both field-programmable gate arrays and complex programmable logic devices, offer a design platform that allows students to work on more meaningful projects with tens of thousands of gates while still learning the fundamentals of digital design [7], [14].

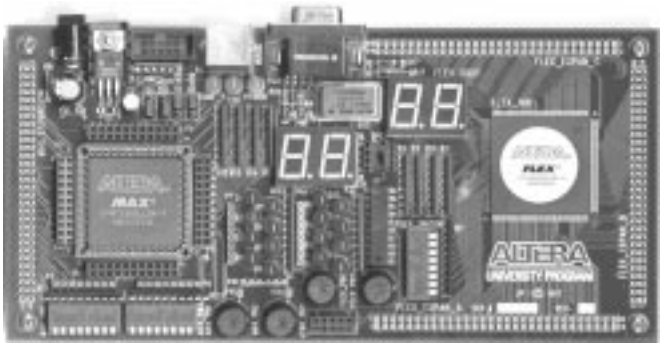

Fig. 1. Altera's UP 1 board contains two FPLDs. All projects discussed in this paper use the larger FLEX 10K20 FPLD located on the right-hand side of this board. This FPLD contains the equivalent of ~20 000 logic gates.

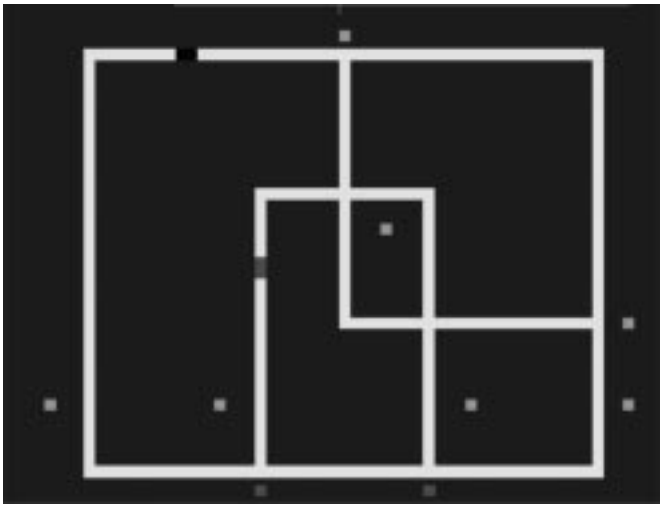

Fig. 2. The train simulator outputs this graphical view of the tracks, trains, sensors, and switches to a VGA monitor via a connector on the UP 1 board. This gives students immediate visual feedback that can be used to test and debug their state machine controllers, which guide the trains around the simulated tracks.

Under the new course structure, ECE 2031 is the introductory digital design laboratory at Georgia Tech. It is a two-semester-hour laboratory course that immediately follows a three-semester-hour lecture course, which introduces students to the traditional, fundamental concepts in digital design and computer architecture. The ECE 2031 lectures are motivated by the laboratory projects and, as such, must avoid excessive abstraction and theoretical analysis in favor of specific instruction on the project at hand.

Manuscript received August 14, 2000; revised February 21, 2002.

K. E. Newman is with the School of Engineering and Computer Science, University of Denver, Denver, CO 80208 USA.

J. O. Hamblen and T. S. Hall are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA.

Publisher Item Identifier S 0018-9359(02)05044-6.

0018-9359/02$17.00 © 2002 IEEE

## II. Laboratory Logistics

The students' workload is divided into three phases: prelab, in-lab, and postlab. Before each laboratory session, students attend a one-hour lecture given by a faculty member. They are also required to complete "prelab exercises" before coming to the laboratory. These exercises vary from week to week, but they are designed to take no more than two hours per week and typically include reading from the textbook, working through online tutorials, and/or completing an initial circuit design using the freely distributed student edition of Altera's MAX+PLUS II software. The laboratory session lasts three hours each week and is supervised by graduate and undergraduate teaching assistants such that there is, at worst, a 6 : 1 ratio of students to teaching assistants. Undergraduate teaching assistants also staff the laboratory during limited hours in the evenings and on weekends for the few students who fail to complete the laboratory assignment during their regular session. After completing each assignment, students must write a laboratory report summarizing their work and results. The length, format, and formality of these reports vary throughout the semester to give students the opportunity of writing different types of reports. On average, students are expected to spend six hours working on each laboratory assignment, including lecture, prelab exercises, and in-lab time.

## III. Digital Design Using CAD Tools and FPGAs

The tradeoffs between using commercial and special-purpose educational tools in a laboratory setting are well debated in engineering educational literature [3]. To provide students with the tools to be productive and innovative in today's rapidly evolving market, academia must keep pace with industry. By using commercial products in a controlled/limited laboratory environment, students are trained to use the tools that will make them immediately productive in industry. They are also provided with standard training in digital design theories and practices that will equip them to challenge current technologies with new discoveries [12]. In 1995, students at Georgia Tech began using commercial computer-aided design (CAD) tools in senior design classes to design, simulate, and synthesize reduced instruction set computer processor cores [4], [15]. Since then, similar CAD tools have been successfully introduced into junior-level computer architecture courses, and now into the sophomore-level digital design laboratory as well.

In recent years, FPLD vendors, such as Altera and Xilinx, have offered student versions of CAD tools for a nominal fee or for free. These vendors also offer reasonably sized FPLDs (currently 4000 to 70 000 equivalent gates) integrated onto student development boards. (Several commercial offerings are highlighted in [2].) In particular, Altera's UP 1 board, shown in Fig. 1, contains two independent FPLDs, pushbuttons, DIP switches, light-emitting diodes (LEDs), a 15-pin high-density D-Sub connector for attaching a VGA monitor, a PS/2 connector, and external I/O headers. The Altera UP 1 board is sufficient for both the introductory digital design laboratory and the senior design laboratory (although the senior design laboratory uses larger FPLDs on the new UP1-x board).

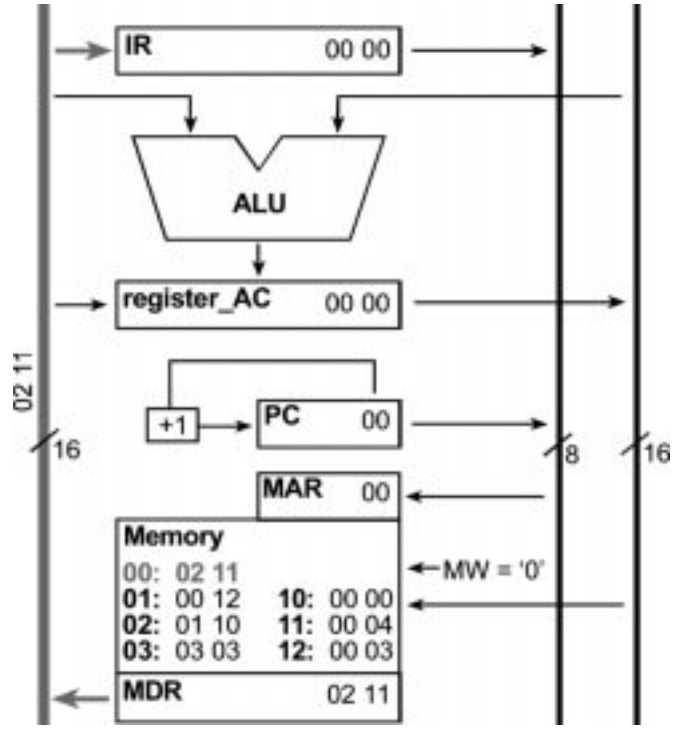


Fig. 3. The simple computer architecture uses a multicycle fetch, decode, and execute sequence, a 256-entry combined instruction/data memory, and a single accumulator [2]. Students complete a VHDL state machine model of this 16-bit computer.

There is a significant learning curve that must be overcome when using a complicated commercial tool, like Altera's MAX+PLUS II, in the introductory digital design laboratory. However, the student must surmount this obstacle at some point in the curriculum. By introducing the students to a small subset of tools in a systematic, controlled manner, they are provided with a comfortable environment to quickly master the tools before concentrating on design concepts and theory. In addition to the text and lecture material, two comprehensive tutorials were developed to introduce the basic tool flow of the MAX+PLUS II software [5]. Once the students have completed these tutorials, they have the fundamental knowledge to design a state machine using schematic entry, to compile and simulate their design, and to test the synthesized implementation on the UP 1 board.

Students are also taught a basic subset of the very high-speed integrated circuit hardware description language (VHDL). Throughout the semester, students progress from writing descriptions of simple combinational logic circuits and basic state machines to using VHDL to synthesize simple 16-bit computer models and sophisticated state machine controllers for an autonomous robot.

Introducing students to VHDL at such an early stage allows later courses to be more aggressive and thorough in their treatment of HDLs since students already have a familiarity with the concepts. Likewise, students are able to develop more advanced projects in their senior design classes because the tools and HDL are familiar to them. Students have reported that this early treatment of VHDL has allowed them to be more productive in cooperative and intern positions and to be more marketable in obtaining full-time employment upon graduation.

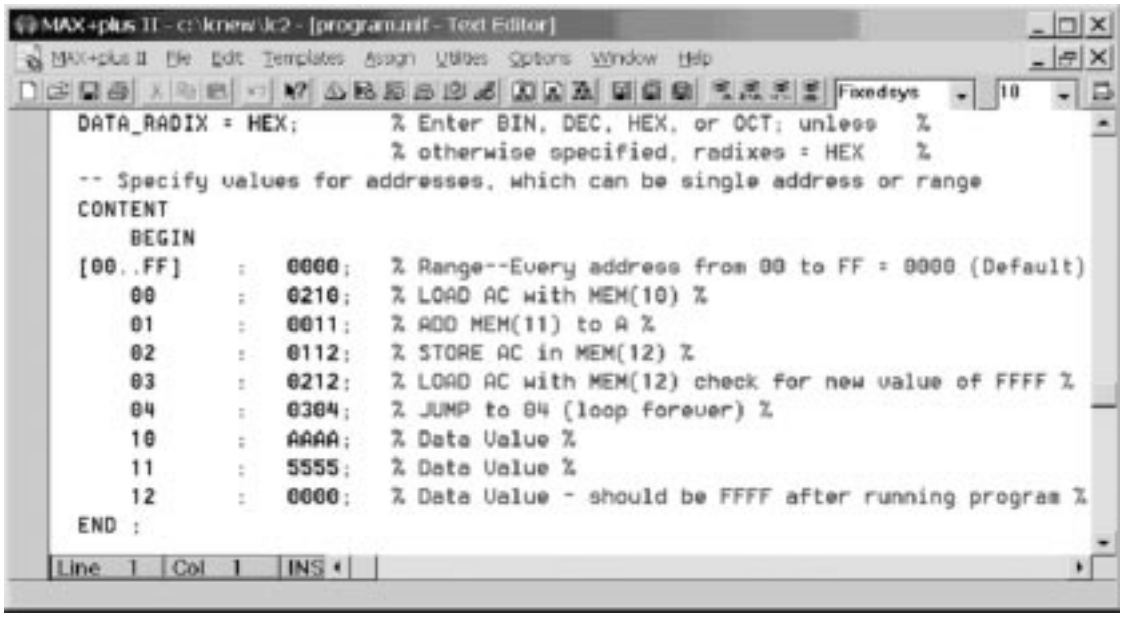


Fig. 4. A sample memory initialization file that contains instructions and data for testing the basic operations of the simple computer.

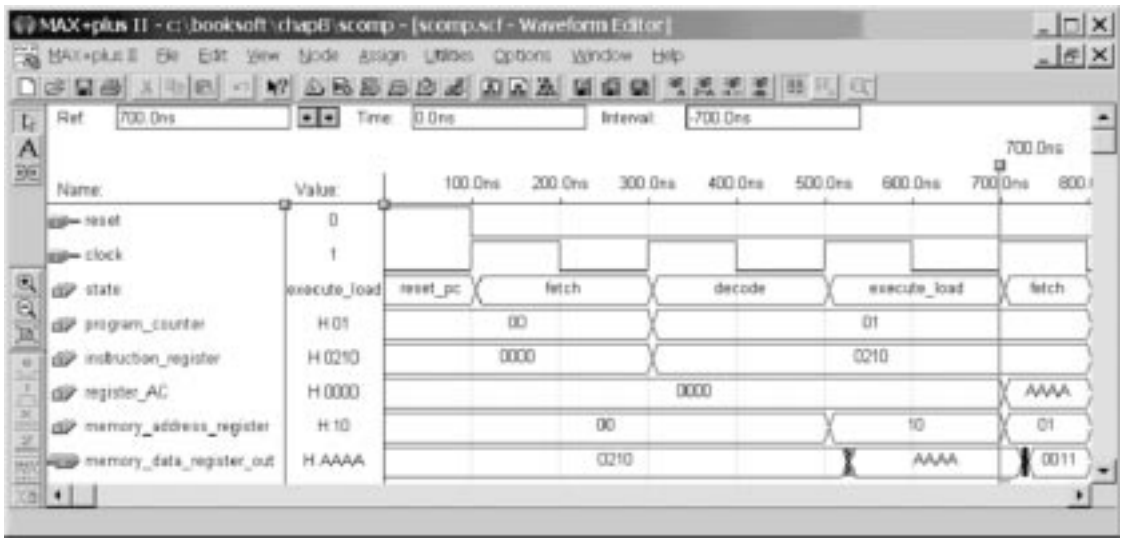


Fig. 5. The waveform output from the MAX+PLUS II simulator when the memory is initialized with the MIF file shown in Fig. 4.

## IV. Laboratory Projects

### A. Traditional Protoboard Approach

While adopting top-down pedagogy for the overall curriculum, the authors did not want to overlook the fundamental concepts within ECE. During the first quarter of the semester, students are introduced to TTL protoboard-based design using 7400-series TTL logic integrated circuits (ICs). These early laboratory assignments are designed to introduce basic logic gates, counters, decoders, multiplexers, classic state machine design, and more. The laboratory assignments in the second quarter of the semester center on instrumentation and circuit characteristics. Students become familiar with digital oscilloscopes and logic analyzers and learn how to measure circuit characteristics, such as rise/fall time, propagation delay, and duty cycle.

### B. State Machines

The treatment of state machines provides the transition between protoboard-based design and FPLD-based design. Initially, students use the schematic entry tool in MAX+PLUS II to design a discrete implementation for the provided state machine. After simulating their design to verify its accuracy, they build it on a protoboard using TTL logic ICs. Finally, they design the same state machine in VHDL and simulate its operation. Simulation waveforms are compared for the two implementations and with the original state diagram to verify the correctness of each implementation. After being introduced to VHDL in this manner, students can easily make direct comparisons between these two design methodologies.

At this point in the semester, the transition from protoboard-based design to FPLD-based design is completed by requiring students to implement a more complicated state

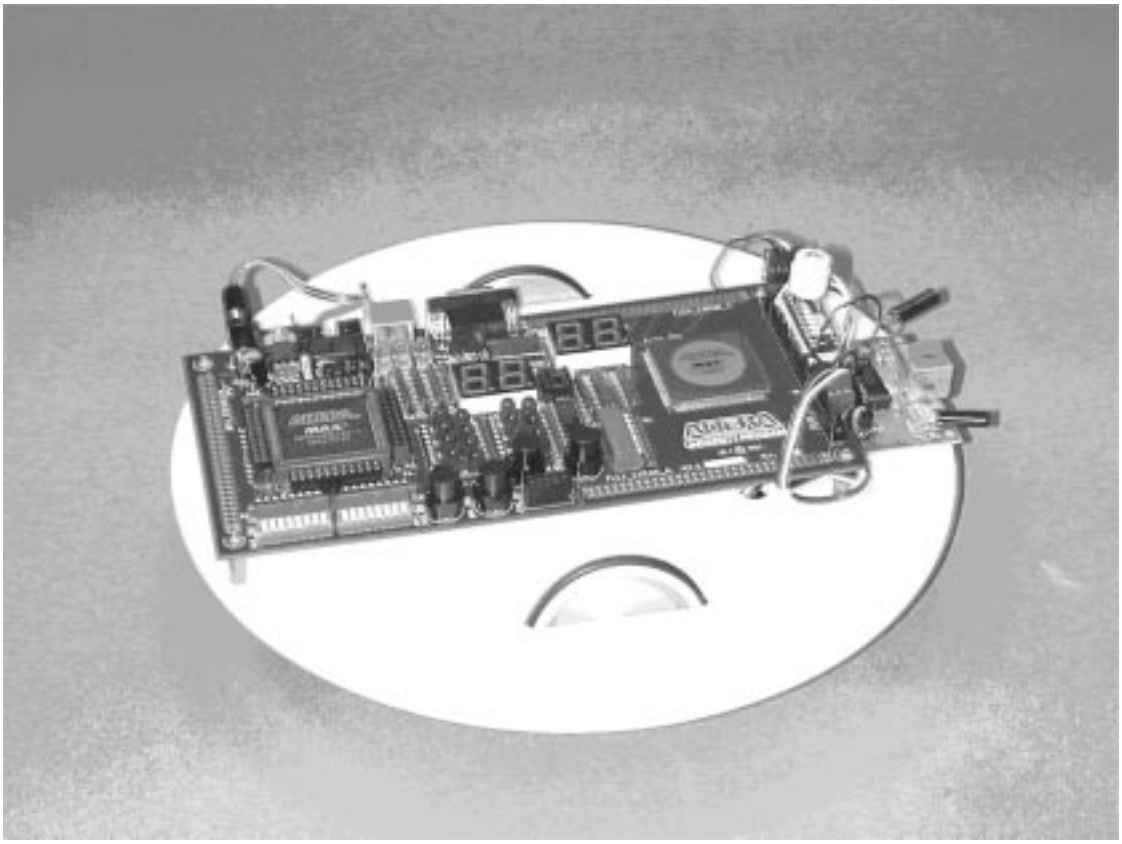

Fig. 6. The UP1bot is controlled with the FLEX 10K20 FPLD on Altera's UP 1 board.

Fig. 7. Some of the sensors currently used with the UP1bot. The digital compass on the upper left is mounted on a PC board designed in-house to interface it directly to the header pins on the UP 1 board. The IR proximity detector with two IR LEDs and one IR receiver is shown on the bottom, and the IR distance detector with offset IR LED and receiver is shown in the upper right.

machine in VHDL alone. This laboratory assignment is written within the framework of an HO-model train control problem. The state machine students write must control a simulation that includes four interconnected tracks, five sensors, two power supplies, four switches, and two model trains. The controller must successfully guide the trains in a defined pattern while avoiding collisions. The train simulator is encapsulated within an intellectual property (IP) core that is given to the students and utilizes the video display capabilities (on a VGA monitor) of the UP 1 board to generate the simulator output that is shown in Fig. 2 [5], [6]. Through this interactive implementation, student interest is kept high.

### C. Simple Computer Synthesis

*1) Architecture:* In the second half of the semester, the emphasis is shifted toward computer architecture. As their first hands-on experience with processor architecture, students complete a state machine model of a 16-bit computer written in VHDL (see Fig. 3). This basic architecture uses a multicycle fetch, decode, and execute sequence; a 256-entry combined instruction/data memory; and a single accumulator [2], [5], [6]. After the students have modeled the required instruction set including Boolean logic, conditional branch, and I/O instructions, they write assembly programs to test their design. Finally, students change the original design to implement a Harvard architecture by separating the instruction and data memory into two distinct memory blocks. This final step not only doubles the memory resources for the simple computer but also requires students to demonstrate a proficiency in the understanding of this architecture. It has been the authors' experience that this final project with the simple computer requires students to have more than a superficial understanding of the architecture.

The simple computer laboratory assignments provide an excellent opportunity to distinguish between the hardware synthesis of the computer model and the software that runs on it. One of the disadvantages of introducing VHDL at such an early

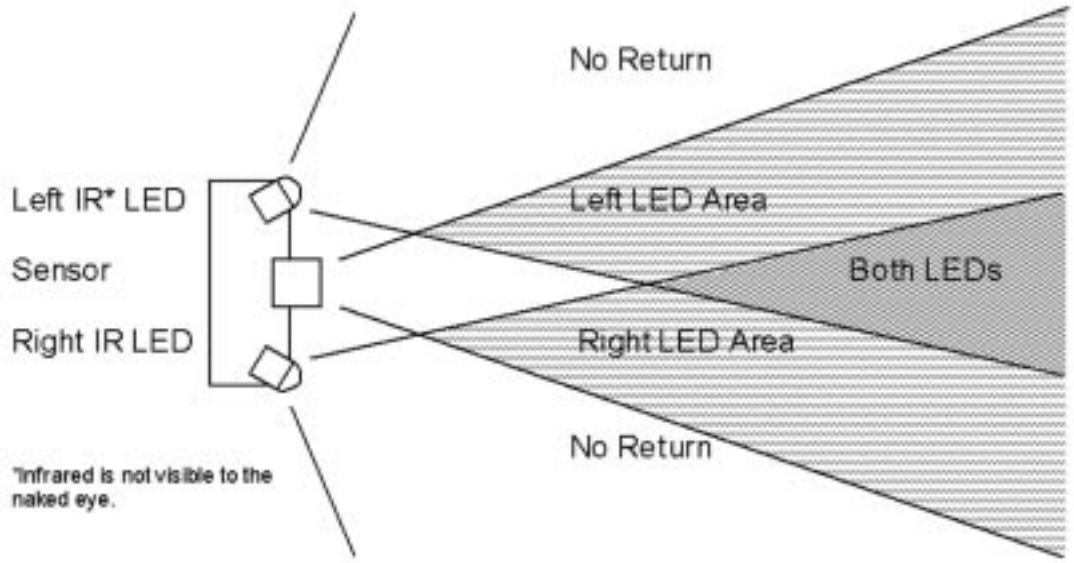


Fig. 8. The IR proximity detector has two infrared LEDs and a central infrared detector with an obstacle detection range of 8 to 26 in from the robot body. By only turning on one LED at a time, obstacles can be detected on the left or the right. If an obstacle is detected in both directions, it is assumed to be directly in front of the sensor.

stage is that students often fail to recognize the difference between describing hardware in an HDL and writing software in a programming language. Great lengths are taken to make this difference clear throughout the semester; however, the simple computer laboratory assignments provide the most explicit example of this difference, since the student is required to design both the hardware and the software.

*2) Modeling:* The simple computer design fits easily into a FLEX 10K20 device. The computer's random-access memory (RAM) is implemented using a RAM module found in Altera's Library of Parameterized Modules. The remainder of the computer model is basically a VHDL-based state machine that implements the fetch, decode, and execute cycles.

The machine language program shown in Fig. 4 is loaded into memory using a memory initialization file (*.mif). This produces 256 words of 16-bit memory for instructions and data. The memory initialization file, program.mif, can be edited to change the loaded program. The computer model is then simulated using the program.mif file as the initial memory values. Fig. 5 shows the output from Altera's MAX+PLUS II simulator. Once the accuracy of the computer model and the instruction/data file has been verified, the design is synthesized and programmed to the FLEX 10K20 FPLD. Students are given an IP core that encapsulates their processor design in a debugging module, which outputs the internal registers of their simple computer to the VGA display [5], [6], [13], [16]. Students step through their programs by clocking the synthesized processor with a pushbutton on the UP 1 board.

## V. Robot Design Project

In the final three weeks of the semester, students work with an autonomous robot like the one shown in Fig. 6, which creates an enthusiasm among the students and provides a valuable learning tool [1]. During the first week, students are introduced to the robot and must complete several simple tasks using a state machine to control movement. The digital compass and infrared (IR) proximity detector are typically used during this introductory week. In the final two weeks of the semester, students work in groups of three or four to complete a design project using the autonomous robot. All design projects are required to use the simple computer students designed earlier in the semester as the robot controller.

Students are provided with three basic sensors for their design project: digital compasses, IR proximity detectors, and IR distance detectors. Depending on the specified goal, teams can use any combination of sensors, including multiple sensors of one type. Although students are given the flexibility to propose their own design projects and use any extra hardware they wish to purchase, experience has shown that this set of sensors proves sufficient for a wide variety of projects, and only occasionally will students bring in additional sensors.

### *A. Robot Platform*

The current UP1bot is equipped with an infrared proximity detector and a digital compass (see Fig. 7). A rechargeable NiCAD battery is attached to the bottom of the platform to provide the robot with power during mobile operation. The robot is constructed using the Altera UP 1 development board [9]. For this project, only the 20 000 gate FLEX 10 K series FPLD is used.

Two modified, low-cost servo motors provide the mobility for the robot, and Teflon skids attached to the NiCAD battery act as a third wheel for the robot. The internal potentiometers on the servo motors are adjusted to a 1.5-ms dead spot and locked in place. Thus, the servo motors are controlled using a pulse-width modulated signal from the FPLD: a 1-ms pulse to turn clockwise and a 2-ms pulse to turn counterclockwise. A detailed parts list and assembly instructions can be found in [5].

When the robot was designed, the two primary goals were simplicity and cost. Simplicity was achieved by not "black-boxing" any component. Whenever possible, sensor and

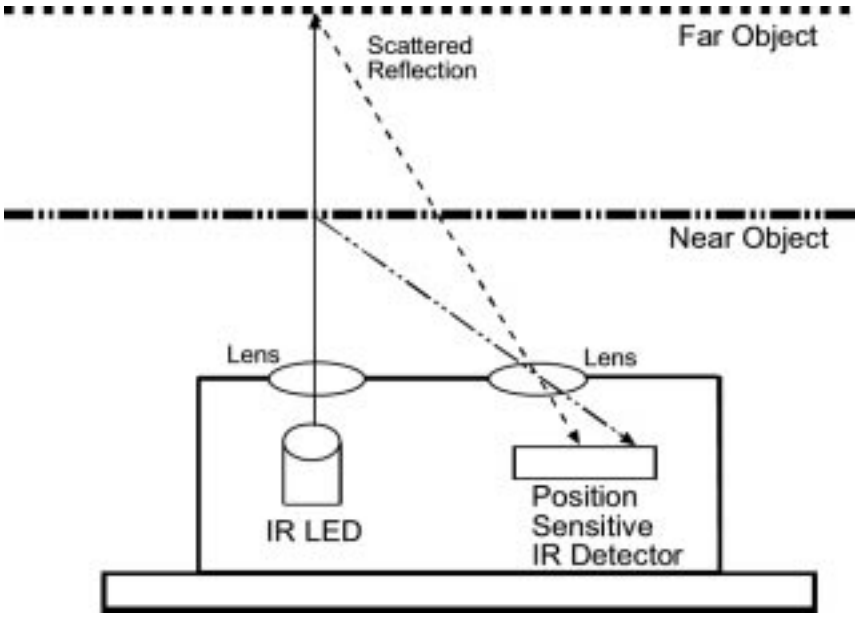


Fig. 9. Operation of sharp IR distance detector.

motor controls are plugged directly into the header on the UP 1 board, and any digital interfacing logic, such as pulse-width modulation for the motor control, is synthesized on the FPLD. The cost of building the robot was kept to about the same as a textbook (excluding the UP 1 board), so that building their own robots was a viable option for students. (Students can purchase the UP 1 board from Altera at a reduced student rate [8].)

### *B. Robot Sensors*

*1) Compass:* The digital compass shown in Fig. 7 is manufactured by Dinsmore Instrument Company (model 1490) and costs approximately $12 [10]. The compass provides eight heading directions by measuring the earth's magnetic field using a hall-effect sensor. The sensor is designed to respond to directional change in a manner similar to a liquid-filled compass. If the compass is displaced 90°, an accurate heading is available in 2.5 s with no overshoot. The compass outputs four active-low directional signals, N, S, E, and W, which can be interfaced to the FPLD using pullup resistors to provide standard TTL signal levels. (NE, SE, SW, and NW are all indicated by two simultaneously low outputs.) A small printed circuit board was designed in-house to distribute power and ground connections to the compass unit, mount the pullup resistors in place, and connect the compass to the UP 1 board's header connector. Except for the pullup resistors, no additional hardware was used to interface the digital compass. Implementation of any signal conditioning and/or reading delays is left to the students as an exercise.

Students are provided with a sample FPLD design that displays the heading of the compass on the UP 1 board's seven-segment display to demonstrate the output of the sensor interface. From this example, students are expected to design their own state machine to move their robot in a predefined pattern using the compass for direction and internal counters for timing/distance since the servo motors are not encoded.

*2) IR Proximity Detector:* The robot is introduced to students with the provision of an example design using an IR proximity sensor available from Lynxmotion for about $30 [11]. As illustrated in Fig. 8, this sensor has two infrared LEDs and a central infrared detector with an obstacle detection range of 8 to 26 in from the robot body. Blinders are attached to the diodes to reduce detection errors (particularly from reflections from the floor). The detector responds to a modulated carrier of 38 kHz that filters background noise such as sunlight and external light fixtures.

The students are provided with a sample state machine, which implements a simple obstacle avoidance pattern. There are five states in the design with an internal timer that is used to determine how far the robot has moved or turned.

*3) IR Distance Detector:* The IR distance detector shown in Fig. 7 is manufactured by Sharp (model GP2D02) and costs approximately $20 [10].

This sensor has a range of 10 to 80 cm (∼4 to 32 in). The distance is output by the sensor on a single pin as a digital 8-bit serial stream.

As shown in Fig. 7, the GP2D02 contains an IR LED and a position-sensitive IR detector. The IR LED transmits a modulated beam of infrared light. When the light strikes an object, most of the light will be reflected back to the LED. Since no surface is a perfect optical reflector, scattering of the IR beam occurs at the surface of the object, and some of the light is reflected back to the position sensitive detector. By comparing the near and far object beams shown in Fig. 9, the position at which the reflected IR beam hits the detector is a function of the reflection angle.

The 8-bit integer value reported by the sensor in centimeters is approximately

$$1000 * \tan^{-1}\left(\frac{1.9}{\text{DISTANCE}}\right) + offset.$$

The 1.9-cm term is the distance between the lenses. The offset is the no-object present value returned by the sensor. This offset constant can vary by as much as 17 between different sensors and has a typical value of 25. A close object reports a larger value, and a distant object reports a smaller value. Objects closer than 10 cm will report an incorrect value and should be avoided by placing the sensor away from the edge of the robot.

### C. Robot Control

The robot is controlled using either a state machine controller or the simple computer designed earlier in the semester. Students are required to design both types; however, the emphasis is placed on the simple computer controller since students are required to use it during their concluding design project.

Students are required to make minimal changes to their simple computer model by adding support for multiple I/O devices (using a memory-mapped I/O space). They are also encouraged to consider the tradeoffs between hardware and software implementations and to modify their simple computer accordingly. In the past, students have reduced their instruction space requirements by adding more advanced instructions to handle indirect memory addressing and decrement-and-jump-if-zero cases. In contrast, other students have optimized their hardware by removing convenient yet unnecessary instructions at the expense of larger code sizes.

Many students find their designs too large for either the FPLD or the instruction space and are forced to reconsider their approach. During the last week of the semester, the student teams optimize their hardware implementations and software algorithms to fit in the given space. In the end, the vast majority of groups succeed in correctly controlling a robot, and the variety of solutions is as diverse as the students themselves.

## VI. Conclusion

In the first digital design laboratory, electrical and computer engineering undergraduate students at Georgia Tech are taken from the traditional TTL protoboard-based approach to digital design through the design of their own simple computer core to a cumulative team design project using a low-cost autonomous robot. They are taken from considering basic logic functions, state machines, and their implementations to exploring the tradeoffs between software and hardware solutions.

The low-cost autonomous robot is an interactive platform that provides the students with a hands-on environment to develop fundamental skills in digital design and implementation. It is used to demonstrate the capabilities of digital design and provide the students with an interesting design project. Robotic laboratory experiments progress through the modification of an obstacle-detecting state machine to an open-ended design project using a simple computer that they develop. At the conclusion of the laboratory, assembly language source code is used to control the robot and combinational and sequential designs developed with sophisticated VHDL synthesis of control logic and computer architecture.

## Acknowledgment

More than 1000 students and several hundred teaching assistants have contributed to this work during the past two years. In particular, the authors would like to acknowledge D. McAlister, M. Sugg, J. Vogel, G. Ruhl, E. Van Heest, and M. Kispet for their help in testing and developing several of the laboratory assignments and tools. J. Hanson, T. Mossadak, and E. Shiflet at Altera provided software, hardware, helpful advice, and encouragement. The authors would also like to acknowledge M. Furman for his leadership in establishing ECE 2031 at Georgia Tech during the first two years of implementation.

## References

[1] R. B. Uribe, L. Haken, and M. C. Loui, "A design laboratory in electrical and computer engineering for freshmen," *IEEE Trans. Educ.*, vol. 37, pp. 194–202, May 1994.

[2] J. O. Hamblen, "Rapid prototyping using field-programmable logic devices," *IEEE Micro.*, vol. 20, pp. 29–37, May/June 2000.

[3] N. L. V. Calazans and F. G. Moraes, "Integrating the teaching of computer organization and architecture with digital hardware design early in undergraduate courses," *IEEE Trans. Educ.*, vol. 44, pp. 109–119, May 2001.

[4] J. O. Hamblen, H. L. Owen, S. Yalamanchili, and B. Dao, "An undergraduate computer engineering rapid systems prototyping design laboratory," *IEEE Trans. Educ.*, vol. 42, pp. 8–14, Feb. 1999.

[5] J. O. Hamblen and M. D. Furman, *Rapid Prototyping of Digital Systems*. Norwell, MA: Kluwer Academic, Aug. 1999, p. 254.

[6] M. D. Furman and T. S. Hall, *Digital Hardware Laboratory Manual*: Erudition, Aug. 2001, p. 204.

[7] M. A. Soderstrand, "Role of FPGA's in undergraduate project courses," in *Proc. 1997 IEEE Int. Conf. Microelectronic Systems Education*, Apr. 1997, pp. 109–110.

[8] Altera Corporation. Design laboratory order form. [Online]. Available: http://www.altera.com/education/univ/unv-students.html.

[9] Altera Corporation. (1997, Aug.) University program design laboratory package. [Online]. Available: http://www.altera.com/education/univ/unv-kits.html, ver 1.

[10] Dinsmore Instrument Company. Model 1490 digital compass module. [Online]. Available: http://www.dinsmoregroup.com/dico/.

[11] Lynxmotion Incorporated. IR proximity detector kit/Part no. IRPD-01. [Online]. Available: http://www.lynxmotion.com.

[12] J. Hamblen, H. Owen, and S. Yalamanchili, "Introduction of rapid systems prototyping into undergraduate computer engineering curricula," in *Frontiers in Education*, vol. 95, Atlanta, GA, Nov. 1995, pp. 2c3.5–8.

[13] J. O. Hamblen, "Using large CPLD's and FPGA's for prototyping and VGA video display generation in computer architecture design laboratories," in *WCAE-4*, Las Vegas, NV, Jan. 1998, pp. 1.A.2–1.A.6.

[14] M. S. Nixon, "On a programmable approach to introducing digital design," *IEEE Trans. Educ.*, vol. 40, pp. 195–206, Aug. 1997.

[15] J. O. Hamblen, "A VHDL synthesis model of the MIPS processor for use in computer architecture laboratories," *IEEE Trans. Educ.*, vol. 40, p. 10, Nov. 1997.

[16] J. O. Hamblen, "Using large CPLD's and FPGA's for prototyping and VGA video display generation in computer architecture design laboratories," *IEEE Tech. Committee Comput. Arch. Newsletter*, July 1998.

**Kimberly E. Newman** (S'95–M'99) received the B.E.E., M.S.E.C.E., and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 1992, 1994, and 1999, respectively.

She is an Assistant Professor in computer engineering at the University of Denver, Denver, CO. Her research interests include microsystems testing, digital logic design, and mixed signal testing. She has published several papers on these topics and received a U.S. Patent for her research in the area of PCB solder bond inspection.

Dr. Newman is an active member of ASEE and IMAPS.

**James O. Hamblen** (S'73–M'76–SM'89) received the B.E.E. degree from the Georgia Institute of Technology (Georgia Tech), Atlanta, the M.S.E.E. degree from Purdue University, West Lafayette, IN, and the Ph.D. degree in electrical engineering from Georgia Tech.

He is an Associate Professor in electrical and computer engineering at Georgia Tech. His current research interests include rapid prototyping, high-speed parallel and VLSI computer architectures, computer-aided design, and reconfigurable computing. Prior to receiving the Ph.D. degree, he was a Systems Analyst for Texas Instruments, Austin, and a Senior Engineer with Martin Marietta, Denver, CO.

**Tyson S. Hall** (S'98) received the B.S.C.M.P.E. and M.S.E.C.E. degrees from Georgia Institute of Technology, Atlanta, in 1999 and 2001, respectively, where he is currently pursuing the Ph.D. degree in electrical and computer engineering.

His current research interests include rapid prototyping of mixed signal systems, cooperative analog/digital signal processing, reconfigurable computing, and embedded systems.