# 1 Project Description

## *Results from Prior NSF Support*

The Principal Investigator has not received NSF support pertaining to undergraduate education in the past five years.

## *Preamble*

Our senior level digital system design sequence targets non-engineering computer science students, using a proven pedagogy and its supporting laboratory environment. Although unique and highly successful, the present laboratory needs to be retargetted to a new technology base, and the syllabus needs revision to include topics such as embedded software. The last such overhaul took place in 1990.

This two-semester course sequence was developed by Franklin P. Prosser and David E. Winkel, who retired in 1998 and 1996. Both of these individuals are award winning educators. They conceived this curriculum in 1973, perfecting it in collaboration throughout their careers. Their textbook, *The Art of Digital Design* [6] is regarded by many to be the best introduction to digital design for computer scientists, as it resolutely emphasizes concepts, principals and methods over electronics and technology.

Indeed, this curriculum evolved at an institution that has no engineering programs, of any kind. Furthermore, it was developed by physical scientists whose approach to academic preparation entails a solid grounding of the student's conceptual development in an observable, measurable physical reality. In such a context, there was a strong need for a laboratory environment in which students with minimal knowledge of electronics can explore logical aspects of design. This need resulted in the *Logic Engine*, a platform that provides the electrical infrastructure for exploring newly learned methods and concepts.

*A personal statement from the Principal Investigator:* I took this course in 1977, as a mathematics transfer student. It opened my eyes to computer science, shaping not only my outlook on the discipline and my career research perspective, but my educational perspective as well. In my opinion, this was and remains the most important course a serious computer science undergraduate can take at Indiana University. I am committed to maintaining its excellence and advancing the world-class pedagogy of Prosser and Winkel. Their “hard nosed” but principled approach, taken together with their exceptional skills as educators, has resulted in a curriculum of national significance, whose importance is rising as computer science education becomes more broadly based.

## 1.1 Goals and Objectives

We are starting with proven pedagogy, a highly refined sequence of two courses, perfected over 25 years. In that 25 years, three major technology upgrades of the laboratory have occurred, the latest being in 1990-92 (under NSF/ILI funding). Each such advancement centers on refinement of a design and prototyping platform called the *Logic Engine*, on which the students construct an integrated sequence of digital systems.

### 1.1.1 Educational Objectives

As discussed in the introduction (Sec. 1), the courses and curriculum central to this proposal constitute an introduction to digital system design for senior level students in an arts-and-sciences institution. In our case, at Indiana University, there are no engineering departments or programs. Typically, this course exposes students with a limited background in logic and no background in electronics to the real elements of digital computation. We believe this course sequence is a national model, especially in institutions with limited offerings in computer engineering.

This does not in any way mean that the course is “watered down,” but that its approach is (computer) science oriented. As they now stand, these are among our most demanding (and rewarding) undergraduate courses. The students’ level of accomplishment compares favorably to any upper division course, offered anywhere. We compensate for the students’ lack of engineering background by teaching a rigorous methodology and by providing a highly stable infrastructure in which to do logical design exercises. Many graduates of this course have become valued professionals, not doing digital design, but working with digital designers. They bring a firm grasp of the underlying concerns and contribute a unusual strength in perspective to their engineering groups.

The course takes a pragmatic but highly principled approach, centering on synchronous design using algorithmic state machines (ASM) notation. The methodology is presented in the highly regarded textbook, *The Art of Digital Design*, written by the courses' originators [6]. Educational objectives include:

1. Introducing the physical basis of digital computation, providing an opportunity for those students who are driven to understand "how and why" (our best students, in other words) to ground their classroom abstractions.

2. Presenting a structured design method. The first semester is focuses on system components and the second on systems composed of many components, including microprogram and software executives.

3. Providing a large, complex, but coherent project allowing students to advance their design and debugging experience, to learn implementation techniques, and to directly experience the value of systematic methods,

4. Reinforcing general aspects of computer science at the digital level of abstraction, expanding the students exposure to aspects such as computer architecture, communication, concurrency, testing, and verification.

### 1.1.2 Specific Goals

The overriding goal is to sustain an up-to-date course and laboratory. Students should be exposed to current technologies and a range of design classes. The specific goals are:

1. Updating the course laboratory, including hardware platforms, software tools and documentation, moving its main technology base from *programmable logic devices* (PLDs) to a family of *field programmable gate arrays* (FPGAs). This goal involves these subtasks:

   (a) Redesign of the *Logic Engine*, our platform for student laboratory projects.

   (b) Software integration with the entailed CAD systems, specifically Xilinx Corporation's *Foundation* package, and the supporting tools of the Logic Engine itself.

   (c) Revision and refinement of the main sequence of laboratory exercises.

2. Expanding the laboratory experience in the second course to include experience with embedded control software.

3. Revising the course syllabus to reflect the laboratory experience, including topics in computer aided design and verification, embedded system design, and others. Resolving of the implied "compression" problems.

4. Writing a third edition of *The Art of Digital Design* and reorganizing the course materials for dissemination.

5. Early dissemination of the work to friendly colleagues.

## 1.2 Detailed Project Plan

This plan covers two upper division undergraduate courses, whose catalog descriptions are given below.

> **B441 Digital Design (4 cr.)** P: C335 or H335. Organization and logic design of digital systems. Course presents a structured design philosophy, emphasizing hardwired and microprogrammed control. Boolean algebra, hardware building blocks, circuit synthesis, microprogramming. In the laboratory, students build, study, and debug a working minicomputer from elementary hardware components. Lecture and laboratory.
>
> **P442 Digital Systems (4 cr.)** P: B441. Elements of computer architecture, construction of hardware systems emphasizing a combination of components to form systems, and applications of general principles of computing to digital implementation. Lecture and laboratory.

### 1.2.1 The Present Situation

With the retirement its originators, the Principal Investigator took full responsibility for the two courses in 1999. This was not a discontinuous transition, as the PI has collaborated with his predecessors continually since 1980, using the Logic Engine to build research demonstrations [4], sitting through the courses again, and through continuing pedagogical interactions over the years.

In AY1999–2000 and AY2000–01, we introduced Xilinx technology in these courses and explored modifications to the laboratory assignments. In these trials, we used the existing Logic Engine to build initial prototypes for the proposed new version (See Photos 1 and 2).

In addition, we prototyped two new laboratory projects for the second course, involving microcontrollers and FPGAs. This work was done using prototyping boards available through Xilinx, and later a custom printed-circuit board (See Photos 4–6).

This ground work has demonstrated the viability of our plan, but has also exposed outstanding pedagogical issues that need to be addressed (Section 1.2.4). We have completed the preliminary design of the Logic Engine and a controller for embedded control labs.

### 1.2.2 The Laboratory and its Equipment

Photo 1 shows the current Logic Engine configured for the Digital Design courses [1]. This board is actually a general purpose prototyping platform, with a $6'' \times 8''$ project area. For the courses, we populate this area with wire-wrap sockets (power and ground pins are soldered) for the main lab sequence. A bank of 128 LEDs, 16 toggle switches, and 12 pushbuttons provide for interaction with and observation of the lab project, described below. Subsystems include solid power and ground planes, and a highly reliable, adjustable clock (12Mhz to single-step), and a AM2910 microprogram sequencer with microprogram memory. There are connectors for RS232 serial communication and a parallel PC port. Software support includes a library of routines for computer controlled test development (setting the switches, reading the lights, stepping the clock, etc.), a microprogram assembler/debugger, and other software facilities.

Photo 6 shows a prototype controller board used in the second course. Its main components include an X3010A FPGA and a PIC16F877 microcontroller. The PIC is a low-end SOC device for deeply embedded control software; in addition to a primitive CPU, it contains serial I/O components, pulse width modulators, digital/analog converters, non-volatile

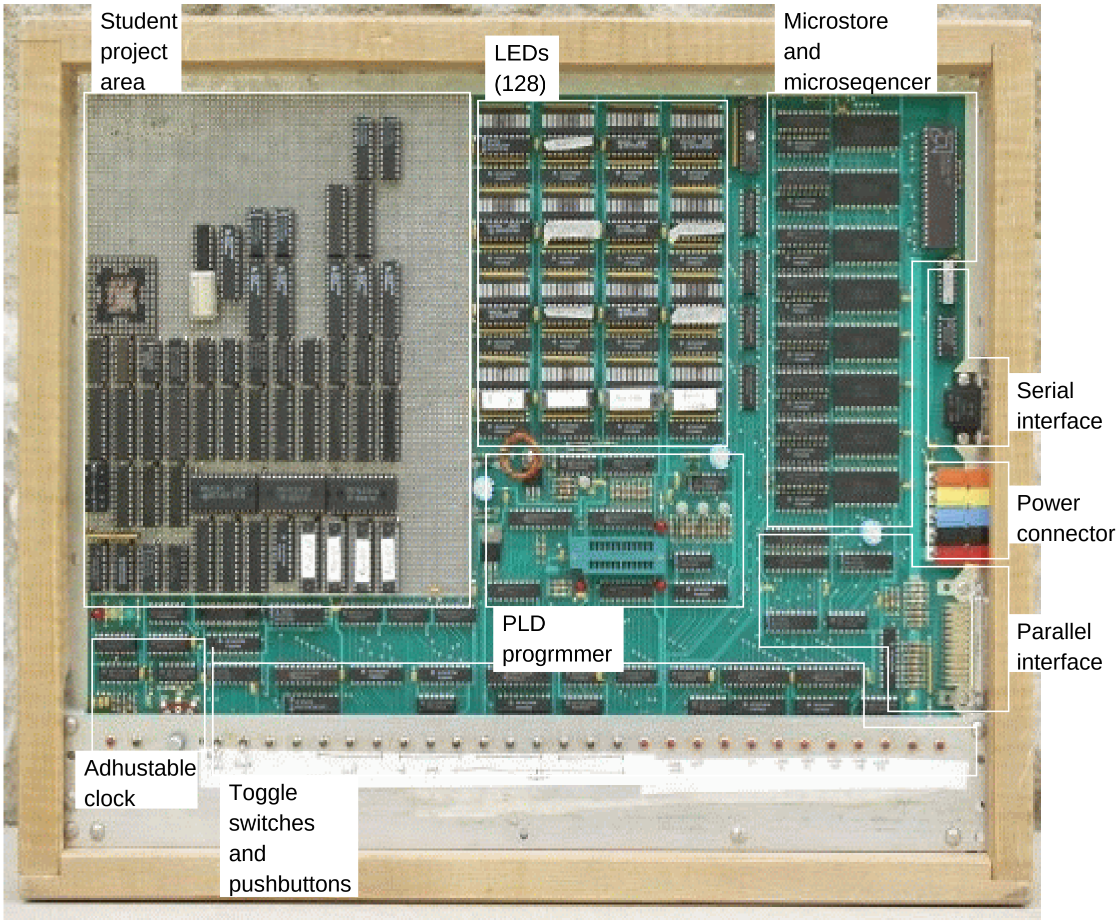


**Photo 1.** Current Logic Engine configured for the Digital Systems Design courses. The dimensions are about $18\frac{1}{2}$"× 14" × 2". The wooden frame has proven to be the ideal enclosure for a device that is handled so much and so roughly. Power comes from a standard PC power supply. The PC development host connects on the right through a parallel port. The project area, top left, hold sockets configured for the PDP8I lab project based mainly on PLD and MSI components.

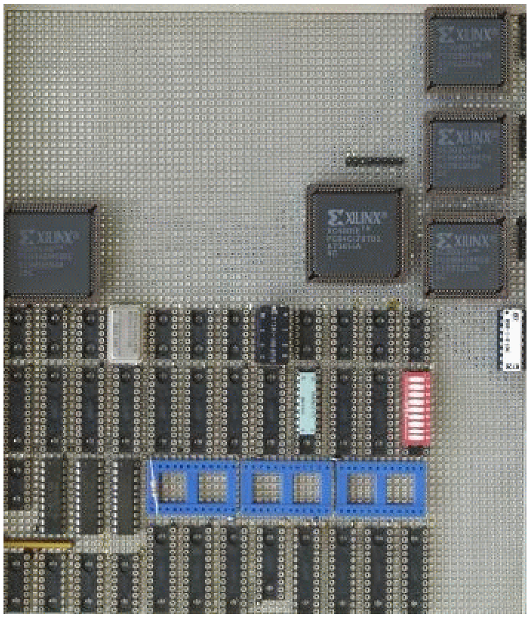


**Photo 2.** This detail shows the Logic Engine project area configured for the Xilinx based lab prototype. Three Xilinx X3020A devices (upper right) contain the data path implementation. The X4010E device to the left of the 3010s contains the "hardwired" control, later replaced by a microsequencer core, microcontrol store, and the students' designs for a UART and PDP8I serial interface.

**Photo 3.** This photograph shows the predecessor to the current Logic Engine, used between 1980 and 1990. It's backpanel is the project area. The base unit holds a primitive M6809 based workstation.

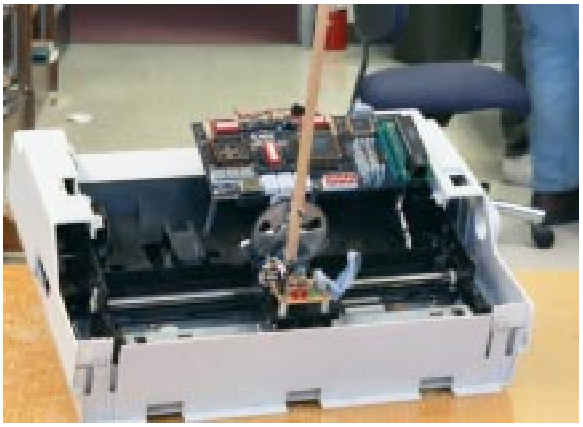
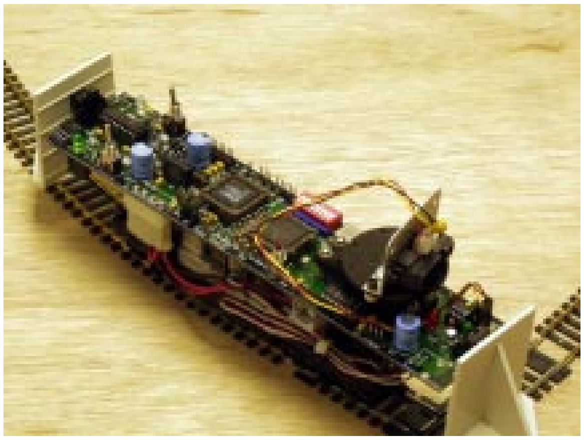
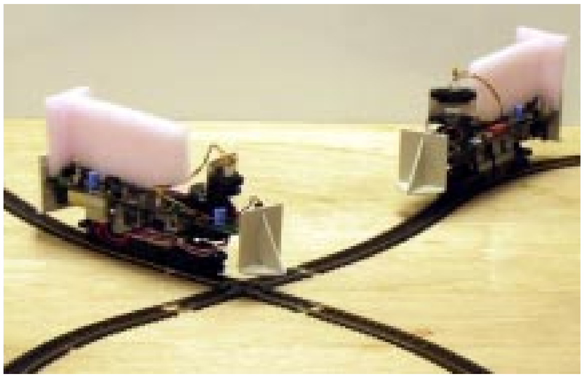


**Photo 4 (left).** Inverted pedulum laboratory exercise. The controller contains a PIC16F877 microcontroller mounted on a Xilinx project board, which also contains an X4010E FPGA.

**Photo 5 (below, left).** The "TrCAS" laboratory project, a collision avoidance system for HO-guage trains. Two or more engines run symmetric autonomous avoidance algorithms on a figure-eight track.

**Photo 6 (below).** The "TrCAS" controller includes a PIC16F877 microcontroller, Xilinx X3010A FPGA, hobby servomotor, infrared range sensor, and power engineering.





program memory, and a RAM.

### 1.2.3 The Laboratory Projects

In the first course, the laboratory consists of 13 assignments [5]. The first two introduce basic instrumentation and the necessary design tools. In Labs 3–12, the students, working in pairs, implement the design of a PDP-8I computer. This is an extremely well conceived project that has proven to be ideal for the course. Its is the right size for a semester of work (See the control ASM in the Appendix), it gives a second exposure to basic instruction processing, and, importantly, it is testable. Lab 13 involves attaching a pre-designed serial data port and interface in order to run Digital Equipment Corporation's original PDP-8I diagnostic instruction tests. Students receive full credit only if their project successfully passes these tests. Students who take this course often say that completing this project was the high point of their undergraduate education.

In the second course there are four projects involving nine laboratory assignments. In Lab 14, students are assigned to replace the UART and interface used in Lab 13 with one of their own design. This lab exposes students to an asynchronous design problem and a level-1&2 communication protocol. In Lab 15, the dedicated controller of the PDP8 data path is replaced by an microprogram, developed symbolically, assembled, downloaded, and debugged using Logic Engine software..

The third project, comprised of three labs, is to balance an inverted pendulum driven by a stepper motor (a salvaged daisy-wheel printer, see Photo 4). The students are *given* a high-level solution to this problem; their task is to implement it using a microcontroller and FPGA [2].

In the fourth project, comprised of four labs, the assigned problem is to implement a collision avoidance system for HO-gauge trains [3]. Again, they have a microcontroller/FPGA combination in which to design and implement a solution. We have deployed the first prototype of a small controller board for this project, and this year's class successfully solved the problem (See Photos 5 and 6).

The last two projects are new. They introduce the new topic of embedded software to the second course. We are still adjusting the details of these projects, with the goal of providing problems in which the better solutions, in terms of effort, involve a meaningful hardware/software decomposition (For example, in both projects motor control should be factored into hardware, among other functions).

### 1.2.4 Outstanding Issues

Design and construction of equipment is in progress, and the laboratory project trials have been successfully completed. There are some substantive pedagogical issues that remain to be resolved. The most important of these are the following:

1. *Imposition of CAD tools.* A major objective in these courses is to give the student direct access to the physical basis of digital computation. Everything is explained, down to VLSI chemistry. The transition to FPGA technology involves a complex architecture and a very substantial CAD environment, imposing a layer of abstraction between the student and the "phenomenon" under study. We simply do not have time in these courses to explain synthesis algorithms.

   Here is an example of the kind of problems we are encountering. In one of our early lab assignments, students design an ALU bit slice using the Xilinx *Foundation's* schematic editor. A strong student's design mapped into 5 common logic blocks (CLBs), more than where intuitively needed. The reason was that the original design contained logic gates with fanout. Once these gates were replicated to eliminate the fanout, Xilinx's mapper reduced the design to two CLBs.

   In many respects, the technology being replaced, 22V10 PLDs, are superior for pedagogical purposes. These devices are essentially physical manifestations of sum-of-products expressions and our programming tool was a fuse map editor. This environment established a very overt relationship between logic minimization skills taught in class, and their impact on implementation in real devices.

2. *Displacement of topics.* Prior to the addition of embedded software projects in the second course, the last project was the construction of a basic computer, including a

system bus, DRAM memory controller, and I/O interfaces. Although this project had substantial value in the student's general understanding of computer architecture, the control labs that replace it introduce equally valuable aspects and technologies that are more likely to be useful. Furthermore, DRAMs and SRAM longer give a sufficient picture of the memory systems they students will actually encounter. In trying to include a general understanding of computer system architecture, we are approaching the limit of student's capacity to absorb material in these courses.

   To compound this problem, we do not give sufficient attention to the concurrency abstractions brought into the course with the introduction of embedded software. This course emphasizes systematic methodology, and it needs to be extended to a foundation for real-time software.

3. *Connection to VLSI* The second course includes short background topics in VLSI chemistry, basic electronics, programmable devices, memories, and asynchronous design. The idea is to briefly present the next lower level of abstraction. This background survey needs to be consolidated to make room for new directions. It requires a delicate balance to provide both substance and necessary breadth of coverage so that students who develop an interest in these lower levels have a springboard from which to pursue them.

### 1.2.5 The Development Time Line

In AY1999-2000 we introduced FPGA technology and conducted a first trial of the main laboratory sequence, using the free area of the current Logic Engine configuration (Photo 2).

In AY2000-2001 we refined the FPGA version of the first course and introduced embedded software projects in the second course (Photos 4–6). We have completed the general design of a new version of the Logic Engine (The Appendix contains parts lists and layouts) and A second embedded control lab was devised and prototyped undergoing a successful trial in the second course.

In the *first year* of the proposed project we will construct and deploy the first version of the new Logic Engine, and make further refinements to the laboratory assignments to address CAD issues. The major remaining task is to develop a microsequencer core that can be incorporated in the FPGA. We are currently using the AM2910 that presently exists on the Logic Engine. The cost of a commercial AM2910 core is prohibitive. A draft third edition of *The Art of Digital Design* will be written to reflect with FPGA technology. After the first course is completed, we will do a final design of the Logic Engine and initiate an effort to revise and consolidate the software libraries and documentation.

In the *second year* of the proposed project, we will construct and deploy the final version of the new Logic Engine and also finalize the embedded control laboratories. The textbook will be completed, and a web page of course resources will be opened. By this time we expect to have enlisted a small number of colleagues at other institutions to plan for deployment of the course elsewhere in the following year.

## 1.3 Experience and Capability of the PI

The Principal Investigator, Steve Johnson, is a active researcher in the area of formal methods for systems. His verification research has centered on hardware, but is now expanding to address embedded software, particularly in control applications. Throughout his career, Johnson has emphasized practical demonstration of his formal approach, building software tools and doing several significant case studies using the Logic Engine as a prototyping platform [4].

Johnson's relationship with the originators of this curriculum goes back to 1977 when he took the course. Winkel and Prosser were on his PhD research committee and were collaborators early in his research career. In addition, they have been his principal mentors in teaching. He has an deep familiarity with and understanding of their pedagogy, including collaboration on three undergraduate courses. He assumed responsibility for the courses involved in this proposal in 1999, when Prosser retired, and has taught them each two times.

Johnson's educational goals are motivated by his research, of course, which requires a background in digital systems in addition to a foundation in programming languages, logic, and verification. His formal perspective on design agrees with (indeed, was inspired by) the perspectives of Winkel and Prosser.

Johnson co-organized the first workshop on formal methods education to be held in the United States (See his biographical sketch). He received a departmental Teaching Excellence Recognition Award in 1999.

## 1.4 Evaluation Plan

This is an EMD Proof-of-Principal project, scheduled for two years. Its main goal is to produce or revise an integrated collection of resources, including a textbook, laboratory manual, and the Logic Engine environment including a hardware platform, software support, and documentation. The ultimate measure of success for this two-year project will be the and organized collection of these materials, successfully deployed in this department and publically available through the Internet. This project will produce an evaluation report in accordance with NSF 93-152 (rev. 2/96).

### 1.4.1 The Engineering Effort

A weekly meeting of the engineers, lab instructor, and the Principal Investigator is already ongoing. This meeting includes a review of the development time line, status report on landmarks, and discussion of arising issues and their resolution. All participants have experience with project management, and some with previous upgrades of this laboratory. Project status will be maintained on the project's web page.

### 1.4.2 Student Assessment

The philosophy and practice of assessment is well established for these courses. However, with the retirement of two highly experienced instructors, and introduction of new topics and projects, assessment practice is under review and refinement. Ultimately, student assessment is primarily based on accomplishment in the laboratory, and standards of achievement have already been established. The main purpose of in-class tests is to give students timely feedback on their progress in mastering those design and analysis techniques needed for success in the lab. However, in this aspect, there is a need for improvement in the second course, which covers a number of background topics (e.g. electronics, VLSI chemistry, etc.) that are not directly applicable in the lab.

### 1.4.3 External Evaluation

The PI will approach his contacts in industry and at other institutions to comment on course content as it relates to student participants and to professional practice. Included will be previous course participants with professional positions relating to these courses. This has been done informally in the past, but for this project, an information packet and questionnaire will be distributed and the results summarized.

## 1.5 Dissemination of Results

In conjunction with the external evaluation just described, articles will be submitted to forums in computer science and microelectronics education describing the course content, underlying philosophy, and resources developed. A project web page will be developed and circulated.

In a 1991 Instructional Laboratory Improvement (ILI) project, this kind of limited advertisement generated enough response for dissemination on a small scale. These courses were adopted at one branch campus, IU South Bend, and the University of Northern British Columbia. In addition, fourteen Logic Engines were built for six institutions and two individuals, but may have been used for courses, research purposes, or both.

The intent of the ILI project was to openly disseminate printed circuit boards and parts lists assuming that small departments, even individual instructors, would choose to do the assembly themselves. In every case, however, built boards were requested and provided at cost ($540). In some cases, an engineer, instructor, or graduate student was tasked to set up the environment during the summer.

We expect this kind of limited deployment to occur in the course of this project, to be followed by a more ambitious deployment effort in the out years. However, this more aggressive effort will require the involvement of a commercial partner.