The Laboratory Environment of the URI Integrated Computer Engineering Design (ICED) Curriculum

Augustus K. Uht and Ying Sun
Department of Electrical and Computer Engineering
University of Rhode Island
Kingston, RI 02881-0805, USA

Abstract - *While many computer engineering curricula contain significant design projects encompassing either hardware or software, most do not integrate these experiences, are short-term in nature, lack exposure to industrial design tools, do not have hands-on experience, or lack continuity. URI's new ICED undergraduate curriculum addresses these issues through a comprehensive multi-year project. The project entails the design and construction of both the hardware and software of a computer, including the compiler, CPU, bus, memory, and I/O. An Ethernet-like network interface is also designed and built, and students' computers are interconnected. Hardware, software, and design-time tradeoffs are examined and made throughout the project. Computer engineering principles come to life in the investigation and design of real and integrated hardware and software.*

This paper describes the laboratory environment of the ICED curriculum. The pedagogical use of industrial CAD logic design and synthesis tools at multiple levels is given, including: state machine, VHDL, and gate levels. The physical hardware used to realize the system's logical design is discussed, including the use of both a high-density Field Programmable Gate Array (FPGA) and discrete digital and analog circuits. Mechanisms are described that employ a logic analyzer with special software to provide correlation between analog signals, digital signals, corresponding assembly code, and high-level source code. Thus, the student is able to intimately investigate the operation and interaction of the CPU, system bus, memory and network, and to understand the impact of hardware, application software and compiler design decisions on computer system performance and cost. The ICED curriculum began in the Fall of 1997.

1 Introduction and Background

The Integrated Computer Engineering Design curriculum at the University of Rhode Island provides students hands-on and in-depth exposure to hardware and software design issues, and the tradeoffs that are made between them[2]. The primary vehicle used to achieve this is an integrated project lasting primarily over the last two years of a student's studies. The project consists of the design, construction and test of a computer, its compiler, and a network interface (the network is comprised of several students' computers). ICED began in the Fall of 1997 for the Junior class.

This paper presents the architecture of the ICED laboratory environment, the major component of the new curriculum. Since ICED is an ongoing long-term project, some detailed design and construction tasks of the laboratory remain to be done.

The laboratory has been designed to support the ICED goals. In addition to teaching principles for a lifetime and design skills for today, ICED integrates a student's core coursework into the project so that the student always has a good idea of why she or he is learning something, and how it fits into the big picture. Further, students are exposed to a truly long-term project, and learn diligence and documentation skills necessary for a successful conclusion. Further, a key goal is to show the student how all levels of hardware and software interact and fit together.

1.1 Prior Work

To our knowledge, there is no other computer engineering curriculum embodying all of the features of ICED; in fact, such a program is rare for any major, including non-engineering majors. One of the closest is that at Georgia Tech[1], where students have been designing CPU's and implementing them in FPGA's for the last several years. Currently, the lab equipment for that short sequence consists of educational prototype boards from the logic vendors. These boards have many capabilities, but lack the flexibility and ease of use of the ICED laboratory. Also see the discussion and references in [2].

1.2 Paper Overview

The laboratory equipment is described in Section 2. In Section 3 the use of the Electronic Design Automation (EDA) tools within the laboratory is presented, as well as how they help to achieve the ICED goals. Section 4 puts things together by describing the progression of the lab work among the core courses. Section 5 discusses both our experience to date with ICED, and some non-technical issues. The paper is summarized in Section 6.

ICED is supported in part by the National Science Foundation through grant # DUE-9751215, and in part with software donations from Mentor Graphics, Xilinx and Virtual Computer Corp.

For ICED details and related papers see: `www.ele.uri.edu/~uht`

2 Laboratory Equipment

The vast majority of the ICED laboratory experiences are built around the design and use of a CPU within an FPGA. An FPGA consists of many uniform logic cells wired together via an interconnection network. The configuration (specific logic functions realized by the cells, and the signal routing and connections) is held in static RAM cells which are loaded *(configured)* when the target circuit is initialized. In commercial products the configuration is normally done at power-on reset time; however, with care an FPGA can be configured at any time, which is a feature ICED uses.

We chose a Xilinx FPGA since they are very flexible and can be used very efficiently. We also chose the Virtual Computer Corp. EVC1 card for the FPGA's platform; see Figure 1. The EVC1 provides us with a very simple interface to the FPGA. Further, by using the EVC1 we were able to save a lot of time and expense, since much of the overhead hardware and support software design and construction was already done.

The students begin their core courses with a typical Sophomore-level introductory digital logic course and lab, currently based on Small and Medium-Scale Integrated circuits. Schematic capture and simulation EDA tools are also used. From the first Junior semester on, the main ICED laboratory setup and EDA tools are used.

2.1 The ICED Prototype System Laboratory

A block diagram of the physical components of the lab is given in Figure 2. Referring to the figure, we now go through the system, describing the major components; bear in mind that the goal is for the student to physically realize an entire computer system.

The major component is the VCC EVC1 card, which is housed in the chassis of a Sun Ultra 1, and is connected to the Sun's SBUS I/O bus. An external connection to the EVC1 is made through the back panel of the Sun via a custom connector/daughtercard assembly called the "d-card." The d-card connects to another custom card, similar to a motherboard, called the "sys-card." The sys-card contains the student's prototype's bus and supporting hardware, including the prototype's main memory. In turn, the sys-card is connected through short cables to a generic protoboard, which holds the non-CPU parts of the student's design work. The sys-card also provides for a standard BNC (Bayonet Nut Couple connector) tap onto the prototype network, ICED-net. Lastly, the HP logic analyzer is used to test, debug, measure and otherwise investigate the prototype. The logic analyzer has 64 data input channels, viewable as state or timing information, as well as analog signal, assembly code, and source code viewing options.

2.2 VCC EVC1 Reconfigurable Computer

The major component of the EVC1 (see Figure 1) is the Xilinx XC4020E-2 FPGA, nominally equivalent to 20,000 simple gates; with an ingenious alternate use of the look-up-tables (normally used to realize combinational logic) as user storage, the effective density is increased.

The EVC1 has many useful features. First, there is a general-purpose interface from the host Sun's SBUS to the FPGA. In fact, part of the interface extends to within the FPGA, and is logically connected to student designs via the EDA tools. Further, many VCC-provided C software utilities are available to the student for such purposes as downloading design files (configuring the FPGA), changing the frequency of a separate clock oscillator (the POM; see the figure), general communication with the FPGA, and resetting and testing the EVC1. The EVC1 also has an FPGA-drivable LED in its back panel; it is used for introductory designs.

2.3 ICED Custom Boards - d-card and sys-card

External connections are made to the EVC1 via its standard daughtercard connectors. These provide access to general I/O pins on the FPGA, configurable as input, output, or bidirectional. 48 signals are sent to the sys-card. On the sys-card, the signals connect to transceivers which are also direction-configurable, this time by jumpers or logic on the student's protoboard.

The sys-card's main elements include the transceivers, as well as the prototype's main memory, a standard SIMM (Single Inline Memory Module). There are also connectors that mate directly with the logic analyzer cables, both for fast hookups and for low distortion. There is also a standard BNC connector used to tie the student's network interface to the ICED-net. This network is 10Base2 Ethernet-like, with similar CSMA/CD operation. Several DIP (Dual Inline Package) connectors are used to communicate with the student's circuits on the protoboard. All of the d-card signals (after the transceivers) go to the DIP connectors, as do all of the signals from the SIMM and the transceiver configuration signals. The sys-card and d-card are custom printed-circuit boards, URI designed.

The protoboard is a generic circuit-prototyping board, composed of several push-in connection arrays. The DIP connectors from the sys-card plug into one of these arrays. There is one protoboard per group (1 group consists of 2 students), kept by the group for the duration. The board holds the interface and memory control circuits, and also acts as a patchboard. It is only attached to the sys-card while the group is actively using the laboratory station during a laboratory session; once the session is over, the group removes its board, allowing other groups to use the station with their own boards.

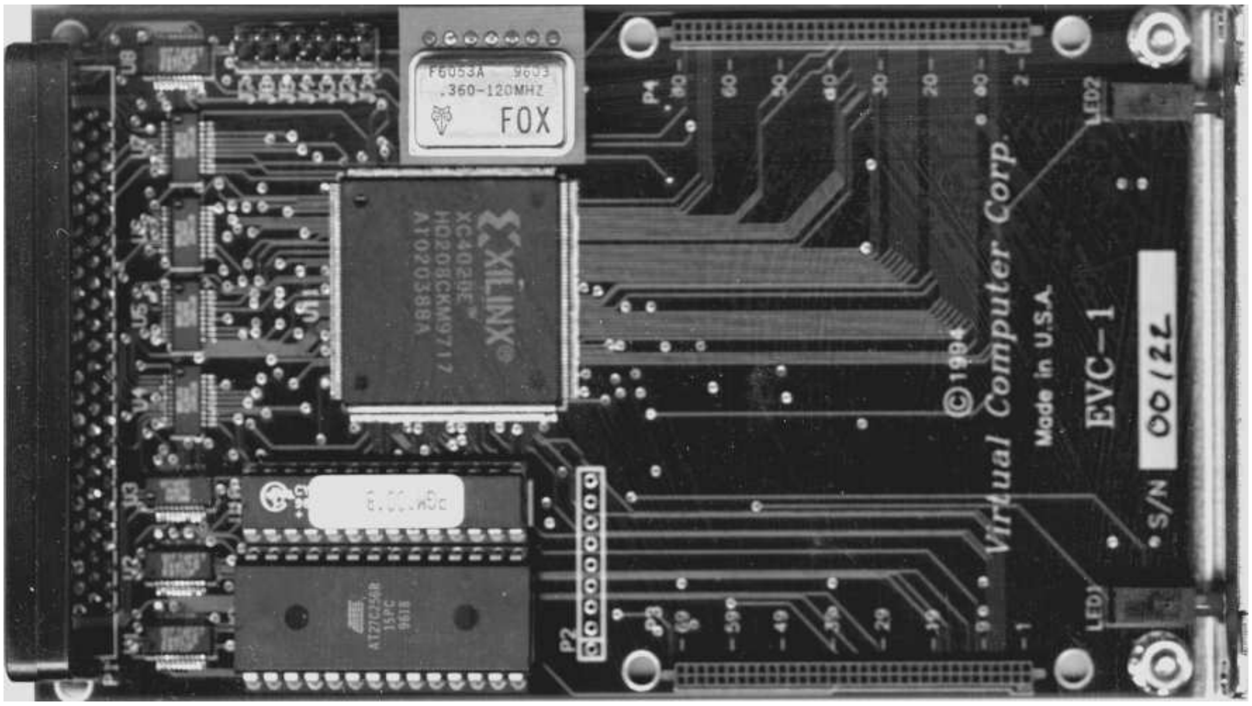

Figure 1: Virtual Computer Corporation EVC1 reconfigurable computer, shown about full size. From left to right, the main components are: Sun SBUS connector (on obverse, beneath handle), interface and FPGA configuration logic, Xilinx XC4020E-2 20,000 gate-equivalent FPGA, POM (Programmable Oscillator Module - marked "FOX"), space and connectors (P3 and P4) for daughtercard (ICED d-card), two pairs of LED's, and back panel plate. A retooled back plate holds a connector providing external access to the d-card, EVC1, and hence the FPGA.

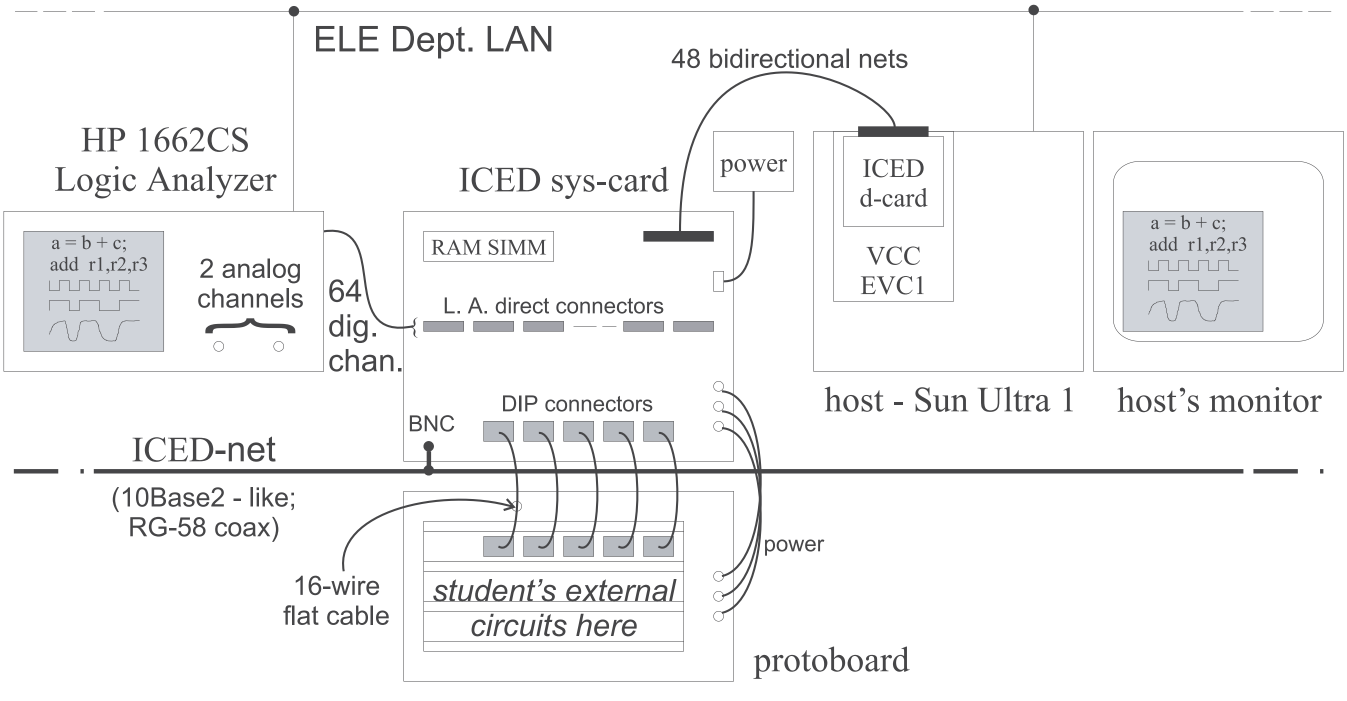

Figure 2: ICED prototype system and lab station. The student CPU is in the FPGA on the VCC EVC1. The sys-card and d-card are custom-made. The logic analyzer is completely viewable-on and controllable-from the host.

2.4 Logic Analyzer

ICED has three complete and one partial lab stations. Each of the complete lab stations has its own high-speed HP 1662CS logic analyzer containing 64 data channels and an integrated 2 channel digitizing oscilloscope. Each analyzer is viewable-on (X-window) and controllable-from its companion Sun host via the department's LAN. The LAN is also used to download disassembly information to the analyzer, as well as to connect the analyzer to a software analyzer running on the host.

3 EDA Tools

A major element of the ICED experience is a large-scale exposure to modern industrial EDA software tools. EDA must be used for complex digital designs; it is impossible to realize such designs without using the tools' automation, simulation and verification capabilities before fabrication. Therefore, our students must be able to use such tools effectively; they are thus included in ICED.

We are using commercial software donated by Mentor Graphics, Exemplar and Xilinx. Although these are complex tools, and sometimes tricky to learn, they are used throughout the curriculum, so the learning time for the student is amortized over many courses.

3.1 Gate-level Tools

The Mentor Design Architect schematic capture program is used for small-scale logic design and simulation in the introductory course. This gives the student a good feel for the reasons behind using EDA tools, especially simulation, and for how the hardware operates in detail, at the gate level. This is important to get across to the student early-on, as it can sometimes become lost in the (later) use of high-level design tools.

ICED employs VHDL, the most common and powerful hardware description language. It is often used as an input to logic synthesis tools, also routinely used. With VHDL, designs can be moved from an implementation on one vendor's chip to another vendor's chip much easier than with proprietary and incompatible design files.

In all but the introductory core course the student uses high-level design tools. Mentor's Renoir program automatically generates synthesizable VHDL from several types of high-level designs, entered via sophisticated Graphical User Interfaces. State machines, truth tables, and block diagrams can be created directly. With Renoir, the student can avoid getting bogged down in most of the circuit and VHDL details, and concentrate on the overall design. However, it does take some practice to become skilled with Renoir (and especially with VHDL).

Modern logic synthesis tools are great time-savers, and greatly reduce logic design drudgery. They have recently become extremely capable; a designer must be highly skilled and have a lot of available time to do a better (low-level) logic design than a synthesis tool. ICED uses the Mentor/Exemplar Galileo Extreme tool for the primary synthesis, and the Xilinx M1 tool suite for FPGA-specific implementation and configuration-data generation. Using these high-level tools is beneficial since it introduces students to modern logic design methods, and allows more time to be spent on register-transfer and higher levels of design, improving productivity. However, there is a significant danger that the student will lose touch with the gate and circuit levels. The tools themselves help with this, since they both have automatic schematic generation and display routines. In Galileo the logic is presented as fairly abstract gates, flip-flops, and modules. The M1 layout editor shows the design's detailed physical wiring and module layout on the FPGA.

The overall ICED EDA design flow is shown in Figure 3. There are many possible flows; the normal path is marked with the bold gray lines. The design is entered via Renoir and converted to VHDL. A functional simulation is then performed on compiled VHDL to verify the design's basic worth (there is no detailed physical timing information available yet). The student iterates back to redesign and simulate until the design is functionally sound. Galileo then synthesizes the VHDL into a netlist. The M1 tool suite maps and place-and-routes the netlist. With the layout known, detailed physical timing delays are estimated by M1 and combined with (back-annotating) a VHDL description which is then simulated. Design iterations ensue again; once past this point, there is a high likelihood that the design will work as desired once put into the FPGA. The configuration data is generated and downloaded to the EVC1's FPGA, and finally the circuit is verified and evaluated with the logic analyzer.

ICED also uses software design aids. In particular, a disassembler and a symbol utility are used within the logic analyzer to interpret and annotate the prototype's bus operation. Individual Assembly instructions can then be recognized on the bus.

A large component of ICED is the writing of a complete compiler for a simple high-level language for the student's prototype computer. Using special HP software running on the host, and using information generated by the compiler during an application's compilation run, the correspondence between high-level source code and machine instructions is shown.

With all of the above, the student is able to correlate all aspects of the computer's execution of a program, from the high-level source code to the Assembly instructions, to the machine instructions, to the digital timing info, to an analog signal view, on both the system bus and the ICED-net. This is very powerful, demonstrating the relationships among the different levels.

ICED EDA Design Flow

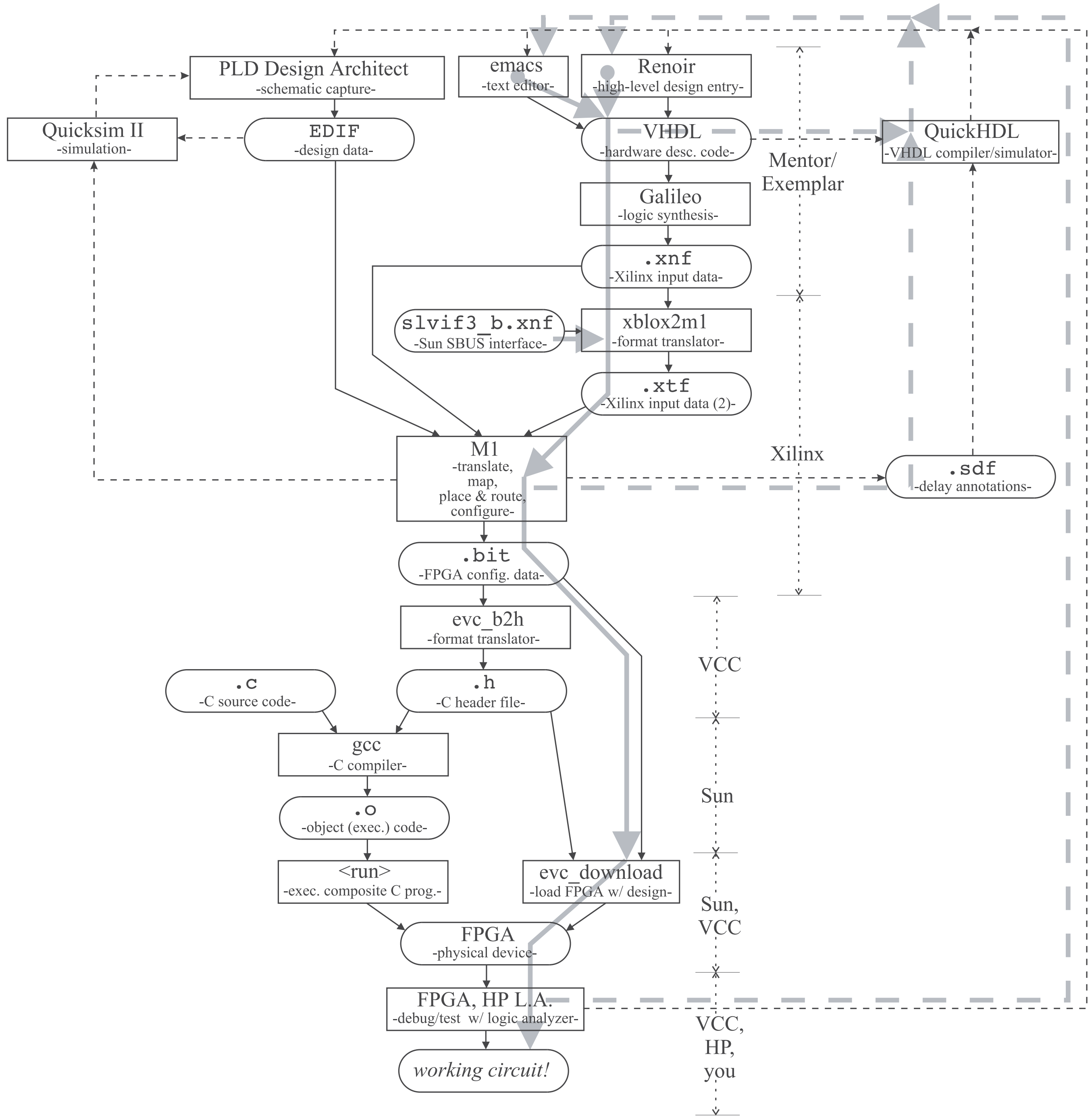

Figure 3: ICED EDA Design Flow. Legend: *-rectangles are operations, ovals are abstract or physical data structures (usually files); -solid lines: primary design flows; -dashed lines: simulation and iteration flows (used for verification and debug); -dotted lines: delimit parts of flow corresponding to the specified software or hardware; -thin lines: possible flows; -thick lines: default ICED flow.*

4 Lab and Course Relationships

We now summarize the overall flow of the design project. The relevant course number and semester taken are in brackets; see the web site for course descriptions.

1. [ELE 201/202 - F Soph] Introductory logic design learned. Introduction to EDA tools.

2. [ELE 305 - F Jr] Prototype design goals set.

3. ["] High-level CPU design done in Renoir. CPU / compiler design tradeoffs discussed. Functional simulation performed.

4. [ELE 405 - S Jr] Simple designs taken all the way through the ICED EDA design flow. Sample design: making EVC1's user LED flash "SOS."

5. ["] CPU with initial bus and small main memory (all in FPGA) realized and tested in physical FPGA. System verified by reading its data back through the SBUS to a host C program. Prototype's clock frequency varied via the POM; this checks the cycle-time design value.

6. ["] External ICED prototype system added, main memory moved to SIMM on sys-card. Logic analyzer used for system verification and performance measurement. Signals viewed in analog, digital and state realms.

7. [ELE 437 - S Jr] Lower ICED-net layers designed.

8. [CSC 402 - F Sr] Compiler written and tested. CPU design modified depending on knowledge gained here, for performance, cost or design time reasons.

9. [ELE 408 - S Sr] Network design implemented and tested. Several students' prototypes connected on ICED-net, overall communication tested. Multi-level code and signal correlations observed. Individual and multi-group system tests and measurements made.

10. ["] Study of effect of design spec changes on software, hardware, viz. cost, design time, performance, etc.

11. [Graduation] Students and professors collapse.

5 Status, Assessment, Reality

ICED has been in place for the Junior class for one year; the feedback has generally been quite positive. In the available survey (for ELE 305) they indicated that they learned a lot. However, the time spent in the two core courses taken (ELE 305 and 405) was excessive. While only one group got their computer to work in hardware by the end of the year, most of the groups had their machines working in simulations. We are addressing these issues by allocating time for the students to catch up next year; the lab environment will be easier to use now that it is less buggy; and the professor teaching 305 and 405 (Uht) will provide a much gentler introduction to the CAD tools, as well as shift some material between 305 and 405. These are growing pains: we will get there.

The students also demonstrated their learning objectively in our on-going ICED evaluation. We administered a test consisting of a design question requiring an essay response to both ICED (ELE 305 [F] & 405 [S]) and non-ICED (ELE 405 [F] & 408 [F]) classes at the ends of the terms. The classes included some EE's and some graduate students; only ICED-Juniors were included in the ICED results. The question was graded blind (1) on the student's ability to make hardware/software design tradeoff decisions, and (2) on the overall quality of the student's response. Each grade was on a scale of 0-10, and each response was graded by two professors. There were no systemic differences in the professors' grading; their grades were averaged. The results are summarized in Table 1. Using a *t*-test with a 95% confidence level, it is seen that the ICED students performed significantly better than the non-ICED students on both scores.

Table 1: Preliminary ICED evaluation scores.

Group	# responses	(1) *tradeoffs*	(2) *overall*
non-ICED	30	3.78 ± 1.28	4.38 ± 1.65
ICED	29	4.48 ± 1.74	5.26 ± 1.80
significant improvement?		**yes**	**yes**

There are several situations in which a student may want to re-start their design: if the student falls too far behind, wants to dump a partner she or he can't stand, works part-time, takes a leave, is a transfer student, etc. This is effectively allowed in ICED at the start of each core course by providing any student with a canned design usable as their own, without penalty.

Since the major lab projects are open-ended designs, significant lab redesign is not often required; when it is, the major components of ICED are readily upgradable, since they are modular or programmable.

6 Summary

The ICED laboratory environment was presented. It is an extensive, flexible, and general testbed for students' hardware and software designs. A student realizes a complete computer: hardware, software, I/O, network, etc., while both seeing classical digital system design principles come to life, and while learning modern design techniques. The student understands the interaction between software and hardware design decisions.

References

[1] J. O. Hamblen. Using Large CPLDs and FPGAs for Prototyping and VGA Video Display Generation in Computer Architecture Design Laboratories. In *4th Annual Workshop on Comp. Arch. Educ., at 4th High Perf. Comp. Arch. Conf. (HPCA-4)*. IEEE, Jan. 1998.

[2] A. K. Uht. The Integrated Computer Engineering Design (ICED) Curriculum. In *4th Annual Workshop on Comp. Arch. Educ., at HPCA-4*. IEEE, Jan. 1998.