A Digital Design Course Sequence for the Computer Engineering Area of Specialization in the Computer Science Department

Afsaneh Minaie
Associate Professor
minaieaf@uvsc.edu
Engineering Department

Reza Sanati-Mehrizy
Associate Professor
sanatire@uvsc.edu
Computing & Networking Sciences Department

Utah Valley State College

Abstract:

In our Computing and Networking Sciences (CNS) Department at Utah Valley State College (UVSC), there are four areas of specialization. The four areas of specializations are computer science, software engineering, networking, and computer engineering. In this curriculum, the students matriculate into the CNS department after successfully completing the requirements of 30 hours of core courses common to all computer science students. The students continue taking core courses until the first semester of their junior year, when they begin choosing their electives from different specialization areas.

Digital logic design courses are fundamental core requirements in both computer engineering as well as computer science departments, in which students get their first exposure to hardware design. It is important that the content of such courses reflect the current design styles used in industry.

This paper describes a three-course sequence in digital logic design that we have developed for our computer engineering area of specialization in computer science department at Utah Valley State College. In this paper, we elaborate the detail content of our three-course sequence and the teaching strategies, and analyze its outcome.

Background Information:

Utah Valley State College is located at Utah Valley, was founded in 1941. UVSC is a state college comprised of two interdependent divisions. The lower division embraces and preserves the philosophy and mission of a comprehensive community college, while the upper division consists of programs leading to baccalaureate degrees in areas of high community demand and interest[1]. Currently, UVSC offers 13 baccalaureate degrees. UVSC is the fastest growing college in the Utah System of Higher Education with over 23,000 students attending. The Bachelor of Science in Computer Science was one of the first Bachelor of Science programs which was approved in 1992 and implemented in 1993.[1]

In addition to the associate degrees, the Computing and Networking Sciences department offers a Bachelor's Degree in Computer Science with four areas of specialization which include Computer Science (traditional), Computer Engineering, Software Engineering and Computer Networking. The curriculum content for the Computer Science degree is based on the 2001 ACM Curriculum Report. The CNS department has 11 full time faculties. Half of the faculties have earned a Ph.D. degree and the other half masters degrees. The Computer Science degree at UVSC is accredited by ABET in 2002 and currently has 869 students. In this curriculum, the students matriculate into the CNS department after successfully completing the requirements of 30 hours of core courses common to all computer science students. The students continue taking core courses until the first semester of their junior year, when they begin choosing their electives from different specialization areas.

The Bachelor of Science in Computer Science program was one of the first Bachelor of Science programs implemented at UVSC in 1993. The program's goal has been to provide a quality program that meets accreditation standards while providing the students with a skill set that allows them to succeed in computing careers.[2] Since our school does not offer a stand alone four years engineering program, the computer science department curriculum contains an area of specialization in computer engineering.

Computing Curriculum – Computer Engineering draft 2004[3] specifies eighteen knowledge areas; sixteen of which relates directly to Computer Engineering and two relate to mathematics (probability and statistics, discrete structures). Comparing Computer Engineering area of specialization curriculum at UVSC with the knowledge areas specified in that draft, it can be seen that our curriculum addresses all of the 18 areas specified.

Introduction:

In industry, there is a big demand for software engineers with hardware background. A project manager for developing a digital system should be able to handle both the hardware and software design issues. "Since the portion of software in a system is getting larger and larger than hardware, software engineers with hardware background will serve as better managers than hardware engineers with software background[8]". Although there is a big demand for software engineers with hardware background, most Computer Science departments still put more emphasis on software curriculum rather than hardware curriculum. Consequently, many software engineers with insufficient hardware background are produced[8].

According to Ivanov[13], there is a well-balanced mix of hardware, software, and theoretical courses in the undergraduate Computer Science curriculum at typical "engineering" colleges. However, at a typical liberal arts college, this balance has shifted significantly towards programming and away from hardware and theory.

In our CNS department at UVSC, we have developed an area of specialization in computer engineering in order to give our students the hardware, software, and theoretical background that is needed to succeed as a computer scientist. In the computer engineering track, students are required to take 17 hours of hardware courses and 6 hours of hardware elective courses[9].

Students who choose this area of specialization can handle well both the hardware and software design issues.

Digital logic design courses are fundamental core requirements in both computer engineering and computer science departments, in which students get their first exposure to hardware design. It is important that the content of such courses reflect the current design styles used in industry. For our Computer Engineering Area of specialization we have developed a three course sequence in Digital Design.

The Three Course Sequence:

All computer science students at UVSC are exposed to digital logic in a freshman course called Computer Architecture and Assembly Language (CNS 1380). In this course, about three weeks (two chapters in their textbook[4]) are allocated for the introduction of combinational and sequential circuits. This course is a core course and all the students in the computer science program are required to take it no matter what area of specialization they are in. Therefore, our students in the Computer Engineering track are exposed to digital logic early in the program during their freshman year.

The second exposure for our Computer Engineering track students comes in a sophomore course called Digital Design I (EENG 2740). EENG 2740 is a four credit hours course with three hours of lecture and two hours of Lab each week. Upon successful completion of this course students should be able to:

- Understand Boolean Algebra
- Learn techniques for minimizing logic functions
- Design combinational circuits such as decoders, multiplexers and adders
- Understand hazards in combinational circuits
- Understand programmable logic devices (PLA, CPLD, FPGA)
- Design sequential circuits such as counters and registers
- Analyze clocked sequential circuits
- Derive state graphs and tables and reduce state tables
- Understand HDL and CAD tools

The overall objective of this course is to provide a general and basic foundation for understanding and designing digital logic systems ranging from simple combinational circuits to sequential designs including finite state machines (FSM). The textbook that is used for this course is Fundamentals of Logic Design by Roth[5].

As the size and complexity of digital systems increases, more CAD tools are being introduced into the hardware design process. The classical (paper-and-pencil) design methods have given way to sophisticated design entry, verification, and synthesis tools. The advent of HDLs, has tremendously improved the prospects for portability of hardware designs. There are two prominent HDLs: VERILOG and VHDL. Both VHDL and VERILOG have the IEEE standards associated with them [10, 11]. Using an HDL, it is possible to describe a design on various levels of abstraction.

It is generally thought that HDLs should be included in a Digital Design courses, and many believe that it should be included at a very early stage in order to promote fluency among students. However, introducing HDLs at a very early stage might be a distraction to the fundamentals of Digital Logic. The first course in digital logic should contain the use of an HDL; however, it should be an introduction that comes after the manual analysis of combinational and sequential circuits. In our Digital Design I course, only the basics of the VHDL Language are introduced toward the end of the semester. These include structural techniques which use components much like schematics and behavioral constructs such as case and if-then-else statements.

Digital Design I course also has a lab component. The goal of the lab assignments is to provide the students with hands-on experience with modern design tools, as well as to support and mature the comprehension of the theory. In the lab, students start with traditional exercises where they build circuits using breadboards and wires. After that they use simulation tools such as Circuitmaker and Altera to design circuits. In keeping up with the goal of providing the students with the exposure and experience with the CAD tools, we have joined the university programs of several CAD vendors. For prototyping of FPGAs and CPLDs, we use development kits supplied by Altera and Xilinx.

The third course in the sequence is the Digital Design II (EENG 3740) which is a junior level course. EENG 3740 is a three hours credit course with three hours of lecture per week. The prerequisite for this course is Digital Design I. This course covers the design and verification of digital systems and emphasizes hierarchical design principles and the use of the Programmable Logic Devices (PLDs). Upon successful completion of this course, students should be able to:

- Define and explain the architecture of various programmable logic devices
- Design combinational logic circuits using programmable logic devices
- Design sequential logic circuits using programmable logic devices
- Design a modern digital system using a hardware description language such as VERIOLG
- Verify correct operation of digital systems using simulation and verification tools
- Understand team dynamics and work effectively on a team
- Understand the purpose and content of technical proposals
- Conduct and participate in professional design reviews
- Write a professional paper and prepare/present a technical presentation

The focus of this course is ASIC design. The hardware design language VERILOG is introduced and used extensively through out the course for designing digital systems and the VERILOG language is treated as a tool. VERILOG takes up about 50 % of the syllabus. The rest of the course is devoted to general design problems: control unit and data path design and timing analysis. The first year this course was taught, VHDL Language was used and after doing a survey we found that VERILOG is widely used in industry. "Over 10,000 industrial designers at such hardware vendors as Sun Microsystems, Apple Computers, and Motorola prefer Verilog[12]." Therefore, we switched to VERILOG. The required textbook for this class is Advanced *Digital Design with VERILOG HDL* by Ciletti [6] and the optional reference is

Fundamentals of Digital Logic with VERILOG by Brown[7]. The students have the option of using three different CAD tools which are Silos II, Xilinx and Altera (Quartus II). The students are given several homework assignments and also four design projects. Toward the end of the semester, the students work on a team project in order to understand team dynamics and work effectively on a team. The students are also required to write a professional paper and present it to class.

The following is an example of a Team Design Project that was given in the Digital Design II class:

- Design an electronic simulator for a Roulette Wheel. Hardware available for the simulator includes 5 push-button switches, 8 slide switches, 4 seven-segment displays, and 24 LED (12 red, 6 yellow, 6 green). The player is allowed to bet on the LED color (red, yellow, green), or LED number (1 – 24). The player may place a bet of $1, $10, or $100. The bet is displayed on the seven-segment display. The amount of money won on each round (based on win/lose, bet and odds) is also displayed on the seven-segment display. While the wheel is spinning the LEDs are sequentially turned on. When the wheel stops, the winning LED is on. The winning LED must be a random event.
- Submit a team proposal describing the functional design of your simulator with major functional blocks and signals identified. Give a short description of the function each block performs and define the inputs and outputs. Define the tasks that need to be accomplished to complete the design along with a time schedule for each task. Define the responsibilities of each team member.
- Submit a team design report including a complete set of documentation with schematics and/or VERILOG code and simulation results.

Conclusion:

This paper described our three course sequence in Digital Design for our Computer Engineering area of specialization in Computer Science department. We feel that our Digital Design courses should emphasize both fundamental issues and technology dependent skills. The fundamentals are basic concepts which are likely to be applicable for several years after graduation of the students and the technology dependent skills which may last only for a few years. Our Computer Engineering track students are getting the fundamentals in all three course sequences and the technology dependent skills in the Digital Design I and Digital Design II.

Any undergraduate Computer Science curriculum should require students to take a well-balanced mix of hardware, software, and theoretical courses. In the CNS department at UVSC, the authors feel that the curriculum for the three out of four areas of specialization has shifted more towards theory and programming, and the computer engineering area of specialization is more balanced in theory, hardware and software. We have developed this area of specialization, in order to give our students the hardware, software and theoretical background that are needed to succeed as a computer scientist.

Higher education institutions abroad may benefit from adopting this three-course sequence in Digital Design in their Computer Science curriculum.

Bibliographic Information:

[1] http://www.uvsc.edu , UVSC – Department of Computing and Networking Sciences, access on December 2003.

[2] Computer Science Program Self Study for the 2002-2003 ABET Accreditation Cycle, Computing and Networking Sciences Department, Utah Valley State College, 2002.

[3] http://www.eng.auburn.edu/ece/ccce , Computing Curricula - Computer Engineering Report (Ironman Draft), access on March 2004.

[4] Warford, Stanely, Computer Systems, Jones and Bartlett Publishers, 2002, ISBN # 0-7637-1633-2.

[5] Roth, Charles, Fundamentals of Logic Design, Thomson, 2004, ISBN # 0-534-37804-8.

[6] Ciletti, Michael, Advanced Digital Design with the VERILOG HDL, Prentice Hall, 2003, ISBN # 0-13-0891614.

[7] Brown, Stephen and Zvonko Vranesic, Fundamentals of Digital Logic with VERILOG Design, McGraw-Hill, 2003, ISBN # 0-07-282315-1.

[8] Chang, Naehyuck and Ikhawn Lee, Embedded System Hardware Design Course Track for CS Students, Proceedings of the 2003 IEEE International Conference on Microelectonics System Education, 2003.

[9] Minaie, Afsaneh and Reza Sanati-Mehrizy, Computer Engineering Area of Specialization in Computer Science Department. Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition, 2004.

[10] IEEE Computer Society: IEEE Standard VHDL Language Reference Manual. IEEE Standards 1076 – 1993, 1993.

[11] IEEE Computer Society: IEEE Standard Verilog Language Reference Manual. IEEE Standards 1364 – 1995, 1995.

[12] Hyde, Daniel, Teaching Design in a Computer Architecture Course, IEEE MICRO, May – June 2000.

[13] Ivanov, Lubomir, Hardware Courses and the Undergraduate Computer Science Curriculum at Small Colleges, CCSC: Eastern Conference, 2002

Biographical Information:

AFSANEH MINAIE is an associate professor in the Engineering Department at Utah Valley State College. She received a B.S., M.S. and Ph.D. all in Electrical Engineering from University of Oklahoma in 1981, 1984 and 1989 respectively. Her current interests are in Computer Architecture, Embedded Systems, Digital Signal Processing, Digital Design, and Computer Interfacing.

REZA SANATI MEHRIZY is an associate professor of the Computing and Networking Sciences Dept. at Utah Valley State College, Orem, Utah. He received his MS and PhD in Computer Science from University of Oklahoma, Norman, Oklahoma. His research focuses on diverse areas such as: Database Design, Data Structures, Artificial Intelligence, Robotics, and Computer Integrated Manufacturing.