Synopsis of CSCE 491: Capstone Computer System Project
2002-2005

Dr. James P. Davis
Associate Professor
Department of Computer Science and Engineering
University of South Carolina

1. Introduction

The CSCE 491 Capstone Computer System Project is a hands-on, project-oriented design course aimed at preparing students for carrying out computer engineering related projects in industry, and has been designed under my direction to expose and immerse the senior CE student in engineering analysis and design work in a problem domain that is challenging, has high value (in terms of its presence on the students' resumes), and has high relevance (in terms of the types of computer engineering problems that will be encountered by students working in this field).

2. Problem Domain

In order to provide a challenging and relevant engineering problem-solving experience, I selected the problem domain of wireless local area networking, using the IEEE 802.11b, "WiFi", standard as the problem domain.

Specifically, the project objective for the students in this course has been to complete a cycle-accurate simulation model of the 802.11b MAC Layer protocol. Developing such system models has been the basis of work I have carried out in industry in recent years. The project represents a "sampling" of a real-world design project that they could be asked to carry out in industry today. We take a subset of the MAC protocol, carry out the analysis of the MAC Layer protocol and its architecture (spending some time on developing this architecture), and realize a design model that is subjected to simulation verification (in terms of functionality and timing). The resultant model can be carried down to actual circuit implementation through the use of automated logic synthesis tools (discussed later).

We have focused on implementing MAC layer functionality to support the 4-frame transaction typically supported in ad-hoc wireless networks—namely, the RTS-CTS-Data-ACK frame set. This set of four frames (and the possibility of framing fragmentation, frame transmission timeout, and framing retries) represents greater than 80% of frame traffic in a typical 802.11b ad-hoc network (operating exclusively in DCF mode).

The assumption has been that the students coming into this course have not had a course on networking, nor have they had any direct design course work related to computer engineering systems. Since the course involves the architectural analysis and design of digital circuits implementing a subset of the 802.11b MAC protocol, the only assumption about the students' knowledge on entering the course is that they have had a Digital Logic course (CSCE 211), a Computer Architecture course (CSCE 212), an Algorithm Design course (CSCE 145), and a Software Engineering course (CSCE 240); knowledge of Embedded Systems (CSCE 313) has proven to be beneficial, but there have been many instances where students have not completed 313 before coming into 491. This set of prerequisites differs from those published in the department's course outline (CSCE 240, 311). I have not pursued to remedy this difference for two reasons: (1) the approach for this course was somewhat experimental in the beginning, and it wasn't clear exactly what knowledge sources the students would need to draw upon, and (2) these courses are required courses for the CE student anyway.

3. Problem-Solving Methods & Tools

This design course is not a programming course, and we focus on the fundamental issues of analysis, modeling and design. To that end, we use two complementary model-based design methods: (1) Unified Modeling Language (UML), and (2) Executable Algorithmic State Machine (ASM). The students come into the 491 course having been exposed to UML in earlier coursework (namely, CSCE 240). Students may have been exposed to the ASM method in their digital logic course (CSCE 211), as the ASM method is featured prominently in most of the leading tests on the subject (and in the Wakerley, 3rd ed., text that we use in 211 at present).

The UML notations are used to explore the 802.11 problem domain, and all concepts associated with the 802.11 domain are represented and discussed using UML diagrams. The following UML notations are used to communicate the protocol and architecture of the 802.11b standard: (1) *Class diagrams* to define important conceptual constructs; (2) *Use Case diagrams* to define an inventory of principal transactions to be supported by the system; (3) *Sequence diagrams* to denote relevant operating scenarios of the architecture, to partition the system into a set of concurrent, interacting components (according to the CRC partitioning method used in object-oriented analysis and design, cf. Wirfs-Bocks, 1991), and to identify the methods of concurrent interaction between partitioned components; (4) *Statechart diagrams* to articulate the state-based lifecycle-oriented behaviors inside relevant system components; and, (5) *Activity charts* to articulate the algorithmic-oriented pseudocode representation of other components in the system. Given the modeling of behavior inside of partitioned components, either Activity charts or Statecharts are used; both are not used together, as they model different perspectives of internal component behavior.

The ASM notation is used to represent the cycle-accurate behavior of the concurrent, interacting components of the MAC Layer model. The ASM notation is a combination of several models into a single notation—namely the finite state machine (FSM) model and the register transfer notation (RTN) model, both of which are used to convey the structure and behavior of digital circuits and systems at the algorithmic and register transfer levels of systems modeling and design. The ASM method allows complex models of digital circuit behavior to be represented as a set of coordinated "threads" of execution that represent the state machines and datapath associated with realizing the 802.11 MAC layer architecture in a custom logic/programmable logic digital circuit. In this sense, we employ the ASM notation and modeling methodology to create an execution model that realizes the protocol directly onto a digital circuit without the use of an intervening microprocessor or its instruction set. Students are able to use the ASM notation to realize a working model of the MAC layer protocol as a digital systems model for subsequent synthesis in a programmable logic device (such as the Xilinx Spartan-IIe FPGA).

Initially, we spent more time in the course working with UML, and therefore, we were using the IBM/Rational Rose tools for creating UML diagrams. For the creation of ASM models, we have been using the flowHDL®(R) tools originally marketed by Knowledge Based Silicon and, subsequently, adopted the Nimbus™ tools marketed by Exsedia (for which the CSE department was granted a yearly renewable license for up to 50 seats of their tools in exchange for user feedback and testimonials on created designs using the tools). The arrangement with Exsedia is similar to that which the department has with IBM for the Rational Rose software (administered by Dr. Vidal).

The *flowHDL* and *Nimbus* tools allow the students to capture their models using editor tools, and to subject their models to cycle-accurate simulation using the tools. This allows the students to devise elaborate debugging and test verification scenarios, and also to create test "harness" models that subject their design models to sequences of fairly sophisticated verification tests. In some instances, student teams have written ASM test "threads" to load randomly generated packet patterns into on-chip memory arrays, and to stream these packets through their MAC models, having additional threads emulating portions of the PHY layer for purposes of sourcing or sinking the frame traffic into and out of the model.

4. Project Experience

The students are organized into project teams of 2, 3 or 4 (depending on the number of students signed up for the class). In some instances, I have allowed individual students to carry out the project work solo (such as for students who wish to use their project work as the subject of their Senior Honors Thesis—for which I have had 4 students do this in the past 2 years).

Generally, the division of workload is based on two students being assigned responsibility for the design of the MAC Receiver and 2 other students the MAC Transmitter. When only a team of 2 is present, I assign them responsibility for either the Transmitter or Receiver, but not both. A single student is assigned the Receiver. A team of 3 students may be assigned both Transmitter and Receiver, but may have some reduction in functionality (generally this entails elimination of the CRC processing or the

---

(R) *flowHDL* is a registered trademark of Knowledge Based Silicon Corporation, *Nimbus* is a trademark of Exsedia, Inc., *Xilinx*, *Spartan-IIe* are registered trademarks of Xilinx, Inc., *Rational Rose* is a registered trademark of IBM Corporation.

frame fragmentation functionality). Special accommodation has also been made in the past for students with disabilities or special needs.

As of Summer 2005, I have taught this project course 6 times. Once it was taught as a CSCE 790 Special Topics course for graduate students during Summer 2002; another time it was taught to a small group of students as a CSCE 498 course in lieu of 491 (as these students were graduating in the summer). During the initial offerings, almost equal emphasis was given to the analysis and design work, in that I focused 1/3 to ½ of the semester's work using the UML notations; this was at a time before the students entering 491 had any appreciable working knowledge of UML. As this UML knowledge has percolated into earlier courses, I have spent less time on teaching UML and more time on the ASM method, allowing the students to get further into the actual design activities.

During earlier runs of the course, I had the students take on only the design of the 802.11 MAC Receiver. As a result of working with students in post-491 Independent Study projects, I was able to identify a reasonable set of extensions to the architecture model, thus extending the course project to also cover the MAC Transmitter as well. This was carried out as a 498 in Summer 2004, and then was instituted into the main course work for Fall 2004.

It has always been the objective to be able to have students carry their designs directly into a VLSI implementation. Given the availability of FPGA devices and test boards, it is feasible to add this to the course. However, the unavailability of funds to purchase hardware, and the logistical difficulty in administering lab hardware (my experience with the 313 course), this has not been added to the course. However, I have been offering CSCE 611 in Spring semesters immediately following the Fall 491 run, so that interested students can bring their 802.11 MAC project model into the 611 course and continue development through functional enhancement and implementation into a working model on FPGA devices. Approximately 5-6 groups have carried this out in the 3 times I have taught CSCE 611 between 2002 and 2005. One team, in particular, extended their implementation with a comparable one of the 802.11 PHY protocol (Receiver side only).

5. Coursework and Deliverables

The coursework starts with modeling a set of digital circuits pedagogically organized to carry the student from their background in digital logic, computer architecture and algorithms to where they can effectively model and design the MAC Layer functionality. The students must create (via graphical model editing tools) a model and simulate models for the following types of circuits: (1) combinational logic 4-function ALU, (2) sequential logic binary up/down counter, (3) shift-add multiplier for basic algorithmic design practice, (4) UART model for exposure to the basic serial communications pattern and the use of concurrency in the modeling of complex coordinated peer-to-peer behaviors across a serial channel. These problems constitute homework problems, and are carried out individually by each student. The students must learn to use the design automation tools and the ASM methodology to accurately and completely represent the functionality and behavior of the assigned digital circuits.

Prior to starting the 802.11 modeling, the students are tested on their knowledge and understanding of the ASM, its use in modeling digital circuit behavior, and the concepts explored in the homework/lab assignments. The effort up to this point takes up about 1/3 of the course. The remaining 2/3rds of the course are devoted to the acquisition of understanding of the 802.11 problem domain and its analysis and modeling using the ASM methodology and tools. In addition, a fair amount of time is spent discussing the verification testing of their models, and the teams must use specific test plan forms in the testing of each component model.

During the project implementation, the teams are assigned a set of blocks to complete during each week. The purpose of the implementation plan is to give the teams a sense of the pace they must follow in order to complete the project. However, I have found that many teams don't follow the prescribed plan, and end up loading up their project efforts towards the end of the semester. I don't penalize them for this, as long as they complete the prescribed workload and follow the specific instructions for the submission of deliverables.

A second exam is given to the students, in order to provide a "foil" for those students who might not be actively participating in the project work. It is always a possibility that team members may attempt to coast through the project work and not make any significant contribution to the team's efforts. Often, the team members will tell me if this is the case; I have had to fail at least one student because he did not participate in the project work. The second exam quizzes the students on the particulars of the project; namely, I ask questions about the inner workings of the 802.11 protocol that only an intimate knowledge of their project team's design model would provide. The weight of the exam scores is such that the student must demonstrate comprehension of the subject matter in this forum to obtain a grade better than a low B. It is often the case that students who have not stayed involved with their project team can risk a final grade of D or F based on the exam outcomes.

The student teams are responsible for creating a complete model of their digital system, a set of test cases and waveforms from the simulations, and other relevant design data as specified in the Deliverable Instructions posted to the course web page. I do not require the students to prepare a formal report in this course. Rather, in addition to the design deliverables, I have them present a set of PowerPoint slides as a narrative of their project, in addition to providing me a demo. I provide a template for the PowerPoint slides, asking for specific information about design assumptions, specifics on how certain engineering problems were solved, and information about the test results and results of performance/throughput analysis.

The teams provide me a demo of their final working project; often, I ask them to show me how their model handles specific anomalous events—to help me to assess the robustness of their designs. In addition, I also routinely schedule time with teams in the lab to work through modeling and design issues, and to check the progress they are making towards their final deliverable (in much the same fashion as a project manager would do in industry).

6. Evaluation of Results and Outcomes

The student teams are responsible for completing the functionality specified in Functional Specification documents that I have written specifically for the 802.11 project. There are 5 parts: 3 for the Receiver and 2 for the Transmitter. These specifications are supplemented with lecture notes handouts posted to the course web page. The students are graded on the following elements: (1) model completeness; (2) model correctness; (3) model robustness (based on scenarios I present during final demo); (4) innovation in problem solving; (5) scope of verification test scenarios; and, (6) quality of the final presentation (presented by the teams as a set of PowerPoint slides). Overall, I have found that the project teams are able to complete the work in a satisfactory manner.

I provide the students with a set of realistic operating parameters (such as inter-frame spacing delays, latency of the transmission medium, etc.) and ask the students to calculate the throughput of their design models. The fundamental question I pose is this: if the theoretical throughput limit of the 802.11g protocol operating in the 2.54 GHz frequency bands is 56 megabits per second (mbit/sec), can their design model keep up? In other words, what is the "headroom" of their model, and how many frames per second can be processed for transmission and reception?

We make some simplifying assumptions about the operating environment, but the students can calculate the cycle times for each type of frame (RTS, CTS, Data and Frag Data, and ACK), and consider the overall throughput given the 4-frame bracketed transaction being the norm. This past Fall, I introduced some assumptions about the frequency of noise bursts on the wireless channel, so that they would have to take into consideration the overhead for frame retries, given how this is handled in the 802.11 MAC's DCF model of operation.

The examinations test the students individually on their understanding of the practice of modeling the behavior of digital circuits and systems, and also on the particulars of the 802.11 MAC protocol and architecture.

In addition, I have the students keep track of the amount of time they spend on each of the activities associated with the project work. This includes time spent on: (1) analysis and architecture, (2) design entry, and (3) verification simulation and test planning. I don't use this information in the grading of the projects; I have been collecting this information for purposes of effort distribution studies in the execution of design work using several different digital design and programming methods. I am currently in the process of publishing the results of this study as part of my ongoing research into computer engineering design methodologies and problem solving methods.