CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: DigiNAU Digital Logic Self-study Tool Kit	
Sponsor Information: NORTHERN ARIZONA UNIVERSITY School of Informatics, Computing, and Cyber Systems	**Xi Zhou,** Assistant Professor of Practice School of Informatics, Computing and Cyber Systems College of Engineering, Informatics, and Applied Science Northern Arizona University xi.zhou@nau.edu 928-523-0902

Project Overview:

Digital Logic is one of the most fundamental courses for students in engineering degree programs ranging from Electrical Engineering to Computer Science, Computer Engineering or even Mechanical & Civil Engineering. The central aim of the course is to introduce students to the fundamentals of basic logic reasoning, binary numbers, and design of digital subsystems – all of which are necessary and foundational for later courses covering the design of microprocessors, other computer hardware, or control systems.

Even though it is a fundamental engineering course, the sudden change from the decimal number system that we use every day to the binary system, in which all values are represented by strings of '0' or '1' -- along with all the new terms and concepts needed to "think outside of the decimal system box", cause a lot of confusion to the students. As an example, one area that causes consistent problems is working with basic logical reasoning tools: translation among truth tables, switching functions, and Karnaugh maps, as well as the simplification of complex switching functions. Due to the limited amount of time in class and the growth in overall student numbers, what is needed is technology that will allow students to independently explore and practice these key skills, receiving feedback from the system to help them grasp the new concepts. In short, what is needed is a "tutoring system", that can support students in practicing and learning logic skills and concepts.

The Problem

This is not a completely new challenge; there are currently quite a few web and mobile apps that can be found online that claim to help students understand digital logic concepts or solve logic problems. The problem is that very few are effective, and all have shortcomings. For instance,

- Some of these apps basically just electronic versions of texts, i.e., they are just restatements of the basic facts that students are already are taught in class. For instance, they might be condensed texts like "Digital Logic Design ABC to XYZ", "Digital Circuits", "Learn Digital Circuits Full" etc. These apps are easy to read, but they don't help the students much in their problem solving skills since they don't provide any practice problems or feedback.

- The majority of the rest apps are simply problem solvers, they can provide students with quick solutions to different kinds of logic problems, such as Boolean function generation, K-map simplification and number conversions. These apps are very useful tools for students who already comprehend the concepts or who are working on projects, but they won't help the students who are learning the skills. Some good representatives include "Boolean Algebra/Kmap solver", "Logic Calculator", "Digital Electronics Calculator" etc.
- There are a few apps that allow students to practice on Kmap or logic circuits, like "Logic Simulator Pro" and "KMAP-Karnaugh", but their functions are limited, e.g., there is no incentive mechanism to keep students interested, they cover only basic cases, etc.

The Envisioned Solution

The goal of this project is to create a multi-platform Digital Logic Self-study Tool Kit (we'll call it "DigiNAU" here; the team can rename it appropriately) where students can practice key digital logic skills on any convenient computing platform, including a lab computer, Android/iOS cell phones and tablets. For this reason, it is suggested that the design be implemented as a web application, with care taken to make it mobile-friendly. Other implementation strategies may be suitable too, and the team will finalize this decision as part of the design process. In any case, the app should include the following key features:

<u>The Basics: minimum viable product</u>

- Can be used across multiple platforms, including Windows, Android and iOS
- Users can easily create accounts, so that the app can remember what learning modules they've tried, and their success rates on them, and can provide users with overviews of progress.
- Allows creation of custom "learning modules" centered around key concepts, e.g., conversion among different number systems; translation among truth table, switching function and Karnaugh map; simplification of switching functions using Karnaugh maps, and simplification of switching functions using Quine-McCluskey method. At least two learning modules must be implemented and demonstrated to prove the concept with this prototype
- Learning modules can be categorized/tagged to distinguish them based on a topic focus, difficulty level, and so on.
- Must provide a "dashboard" page for users in which they can:
- See which learning modules attached to their account and their progress/success on each
- Tracks their overall performance in some way, e.g., gives them a grade for mastery both for individual modules and across all modules they are currently working on.
- Shows a record of their work on each module, i.e., time spent or number of times it was opened and practiced.

<u>A well-appointed product: something that is actually useful</u>

- The learning modules are extensible; new ones can be created and added to DigiNAU later, based on a standard template (or more likely API) that all modules must satisfy. For instance, modules would likely each be centered around some simulator tool (e.g. a number converter) that allows users to explore and experiment with the skill being taught in the module, paired with a set of "challenge exercises" that students should complete correctly, using the simulator element to explore and develop their answers.
- Provides more robust "lessons". Instead of just a series of exercises to do using the simulator/tools in the module, the lessons ask the student to select a "correct" answer to each questions (i.e. multiple choice), and will provide feedback, both if the chosen answer is correct (explains *why* it's correct) or incorrect (explains why).
- Provides a "browser" that, like a browser for movies on Netflix (or similar), allows users to search available modules in DigiNAU; when they find ones, they can "add" these to their accounts and they appear on their dashboard and can be accessed for practicing.

- Provide basic tools to allow faculty to design new lessons (questions/answers/feedback) related to a particular module. In this way, new lessons can be added to the lessons available in some learning module.
- The DigiNAU app interface must be clean and user-friendly, encouraging students to use the system.

Stretch goals: moving towards a true online learning community

- Allow instructors to register accounts as well, in which they can create their own personal "Topic Teachers", which are basically some assigned learning modules plus lessons within them for students to do to learn a particular concept. Thus, to configure a Topic Teacher, instructors can (a) browse for and add certain learning modules available in DigiNAU to it; and (b) within each learning module, "assign" some subset of available lessons associated with the module.
- Teachers can "assign" Topic Teachers as homework for their classes, i.e., by sending them a link by email. Following the link adds the Topic Teacher to the student's account...and time spend, number of attempts, and scores are both recorded for the student and accessible to the teacher on their "dashboard" for that Topic Teacher.
- Teachers can easily "share" Topic Teacher modules, making them available for other teachers on the system to clone into their own accounts, for use with their own students.

If successful as a project, the DigiNAU product will be used as an important supplement teaching tool for the digital logic class; all students taking the class will be required to use the app in the lab or at home. A more distant goal is to make the DigiNAU app available to digital logic teachers anywhere, creating a product that could have truly broad impact in improving Digital Logic education..

Knowledge, skills, and expertise required for this project:

- Knowledge of modern Web2.0 development techniques, and/or Android/iOS development (depending on platform chosen).
- Ability to program effectively in a range of potential implementation languages including C++, Java, Python, and Javascript (again, depending on how the design evolves)
- Basic knowledge about web design, GUI design, and usability principles
- Good understanding of digital logic fundamentals

Equipment Requirements:

- There should be no equipment or software required other than a development platform and software/tools freely available online.

Software and other Deliverables:

- The software product outlined above, installed, tested, and demo'd on a platform of the client's choosing. Must include an excellent user's manual, preferably integrated into the product itself, that walks end users (student learners and faculty) through usage of the product.
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development of the product.
- A systems administration manual, that describes any necessary information for installing, configuring, and maintaining the product. Written for technically competent ITS personnel.
- Complete professionally-documented codebase, delivered both as a repositoty in GitHub, BitBucket, or some other version control repository; and as a physical archive on a USB drive.
-