

# EE382m: Term Projects

Professor Adnan Aziz  
adnan@ece.utexas.edu  
Electrical and Computer Engineering  
The University of Texas at Austin

April 5, 2004

## Introduction

The purpose of these projects is threefold:

1. It is worth 35% of your grade (but this should be the least important item),
2. working on this project should be training on how to go about designing and implementing algorithms for verification, and
3. the project should yield results, in the form of case studies and algorithms that can be made available to the general public.

Projects should be done in individually; I may make exceptions if I feel circumstances warrant them.

## Timeline

I do not want a student going off on a tangent, only for us to learn at the end of the semester that this happened. On the other hand, I don't want to stifle creativity by monitoring things too closely. Most of all, I don't want rush jobs, where everything is crammed into a few days at the end of the semester. You need time to think about these projects.

**Project selection:** You should make a decision as to the projects you are interested in working on by **Monday, April 12**. Feel free to meet with me prior to this (set up appointment by email, or simply drop by).

**Progress reports:** I would like you to provide a one to two page summary on the status of your project on the following days:

1. Wednesday April 21

2. Wednesday April 28

**Project presentations:** Project presentations have been scheduled for Wednesday, May 5. Presentations should be 12 minutes, with 3 additional minutes for questions and answers.

**Final reports:** A 10 to 20 page document (in the format of a conference paper) describing the project should be turned in by 5:00pm Friday, May 7. There will be no penalty for submissions that turned in by 5:00pm Friday, May 15.

## Evaluation

Your grade on the project will be based on a number of factors, particularly the originality and quality of the work. Other considerations include clarity (both the written report and the presentation), and attention to detail.

## Suggested Projects

Below, I give sketches of topics that I would like to see work done on. You are welcome to suggest your own project; however it must meet with my approval. Work which overlaps with your own research area is perfectly acceptable. You can work in groups, but I will expect more from a group than from an individual student. (More than one group may independently work on a project.)

### SymGen with Temporal Constraints

We have seen how Boolean constraints form a natural way to specify environment models. The goal of this project is to enrich the constraints to include temporal constructs. For example, it may be convenient to specify a constraint over multiple cycles.

PLTL is a natural language for specifying temporal constraints; however, care has to be taken to make the semantics work, e.g., I expect it will be problematic to deal with the Until operator.

More than coding up a tool, I would like to see a nice theoretical analysis of what PLTL (or a related language) can express in this context, what its limitations are, what the complexity issues are, etc. Motivating examples would be helpful.

### SAT solving and Symmetry Reduction

There have been huge advances in algorithms for checking the satisfiability of CNF formulas. The goal of this project is to understand these algorithms and see how they can be optimized for designs that contain a lot of symmetry.

## Interface Verification

The goal of this project is to select an interface protocol, e.g., SATA, PCI, USB 2.0, etc., implement it in Verilog, and verify it using Verisity or VIS as a verification tool. You will be responsible for reading through the standards literature and coming up with the design on your own. The deliverables for this project are an implementation of the protocol together with a set of specifications for it, and a document describing your verification strategy.

The write-up should include concrete numbers, such as the size of the largest BDD encountered during reachability analysis, run time, user time, etc. Most such protocols are parametrized by number of processes, datapath width, buffer depth, etc. You should specify the parameters, and show how the verification strategy scales with these parameters.

## Performance analysis

Performance analysis is an integral part of system design. It has been largely carried out in an ad hoc fashion. The goal of this project is to develop some of the theory required to formally specify the performance limits of a design.

The goal of this project is to develop ways in which to do performance analysis for hardware designs. For example, for the tree arbiter, we might be interested in learning what the maximum time is for a processor to receive the token after it has made a request. Of course, this is a function of how long other processors might hold on to the token, in addition to the implementation of the arbiter itself. This particular problem can be solved relatively easily using symbolic model checking. More generally, we might like to know what the average case delay was (again, under some model for the processor). It would be nice to have a logic in which we could pose such questions, and come up with a procedure for answering them. As part of the project, a simple Monte-Carlo style cycle simulator may be implemented, with some mechanism for specifying statistics to be gathered.

## Miscellaneous

A project of this nature will naturally build upon existing work. You are encouraged to build upon existing code/results, and in your report you may copy/adapt from others designs/papers. However, you must explicitly make it clear that you have done so; failure to report this will be considered **plagiarism** and will be dealt with severely.