ECE 697VS: Hardware Verification using Symbolic & Algebraic Computing

Instructor: Priyank Kalla
ECE Dept, Univ. of Utah; Email: kalla@ece.utah.edu; www.ece.utah.edu/~kalla
At UMass for Spring 2017 on Sabbatical Leave from the Univ. of Utah

In hardware verification, it is required to check whether designs satisfy certain properties and behaviours; or that they are correctly designed, where an optimized design implementation is functionally equivalent to a specification model. In order to reason about properties and correctness of the design, formal mathematical models and associated decision procedures have to be devised. This course will introduce students to formal verification techniques – decision procedures, solvers and tools – as applied to combinational and sequential circuit/RTL designs.

Such systems are mostly non-linear; they can be modeled using Boolean and polynomial constraints. Enumerating the solutions to these constraints is infeasible. Modern symbolic computing techniques reason about the solution-sets without actually enumerating them. Boolean verification techniques mostly work on bit-level circuits and models, whereas algebraic techniques can reason at both bit and word-level – naturally possessing the power of abstraction. The course will cover both Boolean and algebraic decision procedures.

Focus of the course: A third of the course will be devoted to verification at the Boolean level using decision diagrams and SAT solvers. The remaining two-thirds of the course will cover hardware verification using computational commutative algebra and algebraic geometry, focusing on *Ideals, Varieties* and *Gröbner bases*. While algebraic geometry is a topic in its own right, we will only consider its application to hardware designs. As hardware designs operate over a finite set of inputs, we will target Gröbner basis reasoning with *elimination theory and Nullstellensatz* over finite fields. Algorithmic focus on domain-specific heuristics to improve hardware verification.

Instruction: Lectures will cover the requisite theory and algorithm design. We will make use of verification and computer algebra tools available in public domain. As part of homework assignments, students will make use of data-structures within these tools to implement verification algorithms, design the circuits and verify them with their implementations. In addition, students will be required to complete a class project. No exams!

The course should be accessible to both ECE and CS students. No prior knowledge of algebraic geometry or hardware verification is needed.