

## Integration of Model Checking and Theorem Proving

Model checkers are specialised proof tools often applied to circuit analysis.

Model checkers can:

- Can solve many circuit correctness problems automatically.
- Are commonly used in industrial VLSI design.

How can we use theorem provers with model checkers:

- To check correctness when model checkers fail.
- To combine the results produced by model checkers into larger results.

## Integration of Theorem Proving and CAD tools

VLSI CAD tools integrate circuit design, analysis, simulations and compilation.

Hardware verification has significant promise, but ubiquitous use awaits integration with CAD.

- How can we make proof tools simple enough for people to use.
- How are the graphical and logical views of a circuit related?
- Can we make logical manipulations graphically?

LAMBDA system from Abstract Hardware Limited was a step in this direction.

## Advanced/Current Topics in Software

- Pointers and Dynamic Memory Management
  - These complicate the semantics and make reasoning difficult.
- Semantics for Object-Orientation
  - Inheritance is poorly understood.
  - Simple semantics and powerful proof rules are current research topics.
- Software for real-time systems
  - Verification of real-time code behavior is difficult.
  - Most safety-critical applications are real-time embedded systems.

## The Hardware/Software Interface

- Verified software can be 'corrupted' by a bad compiler.
- Transformations by optimising compilers are a common source of errors.
- Post-RISC architectures depend on compiler optimisation for performance.

## Advanced/Current Topics in Security

- Security properties of mobile code
  - Can mobile code be proved secure?
  - Can proofs be distributed with code?
- Key exchange protocols
  - Can a key exchange protocol be exploited to discover a key?

## Program Extraction from Constructive Proof

- Constructive logics eschew the axiom of choice and excluded middle.
- To prove that there exists an object in a constructive logic, you must produce a witness.
- There is an isomorphism between constructive logic proofs and terms in the  $\lambda$ -calculus, a Turing complete programming language.

Is doing constructive proofs a viable alternative method of writing programs?

## So, you want to be in formal methods.

- What courses should I study?
  - Algebra
  - Logic
  - Set theory
  - Algorithms
  - Logic programming
  - Functional programming
  - Semantics
  - An application area
- Where should I go for further study?
  - ANU
  - UQ
  - UniSA
  - UNSW
  - Cambridge
  - Edinburgh
  - Glasgow
  - TU München
  - Cornell
  - Oregon Grad. Inst.
  - UPenn
  - UT Austin
- Where can I work afterwards?
  - University
  - Defence Research
  - Defence Contract
  - VLSI and CAD Design