Syllabus

- Go over syllabus
- Introductions
Motivation

Instead of debugging a program, one should prove that it meets its specifications, and this proof should be checked by a computer program.

— John McCarthy
“A Basis for a Mathematical Theory of Computation,” 1961
Boyer-Moore Theorem Provers

- **1970’s**
  - Edinburgh Pure Lisp Theorem Prover (1973)
  - A Computational Logic (1978)
- **1980’s**
  - NQTHM (1981)
- **1990’s-Present**
  - Kaufman joins as developer
  - Workshops (7 already); huge regression suite
- **2000’s**
  - ACL2 books
  - Development environments (ACL2 Sedan, Dracula)
  - 2005 ACM Software System Award (Boyer, Kaufmann, Moore)
Boyer-Moore Theorems Proved

- **1970’s: Simple List Processing**
  - Associativity of append
  - Prime factorizations are unique

- **1980’s: Academic Math & CS**
  - Invertibility of RSA
  - Undecidability of halting problem
  - Gödel’s First Incompleteness Theorem
  - Gauss’ Law of Quadratic Reciprocity
  - CLI Stack:
    - Microprocessor
    - Assembler-linker-loader, Compiler, OS
    - High-level language
CLI Stack

Procedure Mult(var var K: int := 0;
    loop
    if K le 0

formal models related by mechanically checked proofs

0111010100011
0010010000011
011101001001111
0011101001001010
010111011110011

Micro-Gypsy

Piton assembly language

FM9001 machine code

Formal NDL netlist

INPUTS A, B, C:
OUTPUTS SUM, CARR LEVEL FUNCTION;
DEFINE
T0(SUM1, CARRY1) = H
(SUM, CARRY2) =

die plot produced by LSI Logic, Inc, from verified NDL via conventional CAD tools
1990’s: Industrial Applications

- F DIV AMD Floating Point ...
- Motorola CAP DSP
  - Bit/cycle-accurate model
  - Run faster than SPW model
  - Proved correctness of pipeline hazard detection in microcode
- Verified microcode programs
- Rockwell Collins JEM1
- Rockwell Collins AAMP7
  - MILS EAL-7 certification from NSA for their crypto processor
  - Verified separation kernel
Hardware Verification: Motivation


Verification has become the dominant cost in the design process. In current projects, verification engineers outnumber designers, with this ratio reaching two or three to one ....

Without major breakthroughs, verification will be a non-scalable, show-stopping barrier to further progress in the semiconductor industry.

The overall trend from which these breakthroughs will emerge is the shift from ad hoc verification methods to more structured, formal processes.
Hardware Verification Challenge

- Verification costs range from 30%-70% of design cost
- R&D for high-end CPU: 900+ team, costing ~ $1B
- Bob Bently: FDIV bug would cost $12B in 2005 terms
- The verification problem is getting worse
  - Nanotechnology: many inherently unreliable components
  - Multicores: concurrency, coherence, parallelism
ACL2s

- ACL2 theorem prover
  - Runs like a well-tuned race car in the hands of an expert
  - Unfortunately, novices don’t have the same experience
  - Disseminate: wrote a book
  - Not enough: undergrads
- ACL2s: The ACL2 Sedan
  - From race car to sedan
  - Self-teaching
  - Control a machine that is thinking about other machines
  - Visualize what ACL2 is doing
  - Levels & termination
  - Used in several classes
  - We’ll use it in this class
ACL2 is ...

- A programming language:
  - Applicative, functional subset of Lisp
  - Compilable and executable
  - Untyped, first-order
ACL2 Universe

- The ACL2 universe, \( U \) consists of the following objects
- Atoms
  - Numbers includes integers, rationals, and complex rationals
    - Examples include -1, 3/2, and \( #c(-1\ 2) \)
  - Characters represent the ASCII characters
    - Examples include \( \#\backslash2, \#\backslash a, \text{ and } \#\backslash Space \)
  - Strings are finite sequences of characters
    - An example is "Hello World!"
  - Symbols consist of two strings:
    - a package name and a symbol name
    - For example, the symbol FOO::BAR has package name "FOO" and symbol name "BAR"
Conses are ordered pairs (binary trees) of objects

- The left component of a cons is called the car
- The right component is called the cdr

- (1 . "A")
- (1 2 3)
- ((A . 1) (B . 2) (C . 3))

Notes
- The symbols t and nil are used to denote true and false
- Nil also denotes the empty list
- Conses can be used to represent records and finite functions
ACL2 is ...

- A programming language:
  - Applicative, functional subset of Lisp
  - Compilable and executable
  - Untyped, first-order

- DEMO
- ACL2 Web page
- ACL2 Sedan Web page
ACL2 is ...

- **A programming language:**
  - Applicative, functional subset of Lisp
  - Compilable and executable
  - Untyped, first-order

- **A mathematical logic:**
  - First-order predicate calculus
  - With equality, induction, recursive definitions
  - Ordinals up to $\varepsilon_0$ (termination & induction)
Definitional Principle

- Functions are defined using defun
- Example: (defun succ (x) (+ x 1))
- The form of a defun is (defun f ... (x₁ ... xₙ ) body), where:
  - x₁ ... xₙ are distinct variable symbols
  - the free variables in body are in x₁ ... xₙ
  - ... are for documentation and declarations (optional)
  - body is a valid expression (history)
  - if f is recursive we must prove that it terminates
- ACL2 logic is extended with the axiom (f x₁ ... xₙ) = body