

# Trusted Computing in 2025 ?

## UT Austin ACL2-Group Analysis Tools and Capabilities

Natahlie Beavers	Cuong Kim Chau	Soumava Ghosh
Shilpi Goel	Marijn J. H. Heule	Warren A. Hunt, Jr.
Matt Kaufmann	Keshav Kini	Robert B. Krug
J Strother Moore	Ben Selfridge	Nathan Wetzler

**Spring, 2014**

Computer Science Department  
1 University Way, M/S C0500  
University of Texas  
Austin, TX 78712-0233

hunt@cs.utexas.edu  
TEL: +1 512 471 9748  
FAX: +1 512 471 8885

To enable the modeling and analysis of industrial-sized systems:

- Centaur is verifying a contemporary (VIA Nano) x86 design.
- We have developed an x86 ISA model suitable for code analysis.
- We are verifying x86 user-level binary programs.
- We are vetting our tools on commercial-sized problems.
- We are not verifying contemporary OSs and system software yet.

**Question:** Is it reasonable to think that by 2025, we can rigorously analyze a complete computer system's **functional**, **information-flow**, and **security** properties?

Note: Others (e.g., NICTA L4 Micro-Kernel Verification Project, x86 Effort at Harvard, etc.) are also working in this area. For this brief talk, we confine our comments to our efforts.

We have significant collaboration with industry.



**Raytheon**

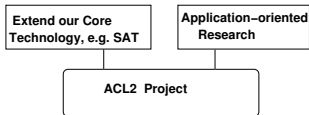
**ORACLE**



**Rockwell  
Collins**

Customers

Our Program

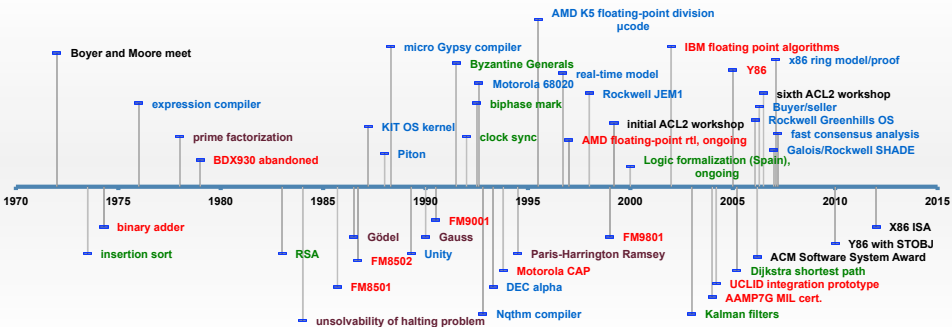


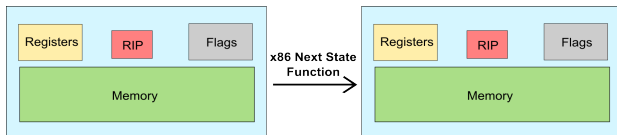
Our own research includes:

- **Development** of core technologies
- **Application** of these technologies on different verification domains
- **Commercial Driver:** validation for Centaur's x86 design

# Timeline

- Our group has been working on the development and deployment of reasoning systems for 40+ years.





We are developing a **formal** and **executable** model of the x86 ISA. Applications of such a model include:

- x86 system emulation
- Binary program verification
- Development of trustworthy programs
- Detecting and analyzing malware and viruses
- Analysis of x86 hardware and software properties
- Build-to (for x86 vendors) and compile-to (for user) specification

Centaur uses our tools to help assure product quality; our tools have displaced a number of commercial tools.

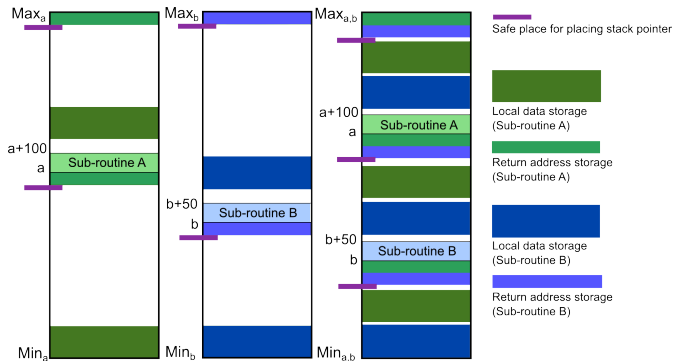
- Centaur models 700,000 lines of Verilog using ACL2
- 3300+ definitions, 4100+ theorems, and 2100+ library references
- Low estimates: macros are heavily used to create such events

Our x86 ISA model can serve as a hardware, build-to specification and as a software, compile-to specification.

- We use our evolving x86 model to verify x86 binary programs
- 400+ definitions, 1500+ theorems (low estimates)
- We mechanically verify small x86 binary programs

# Example: Verification of a Simple WC Program

- Simple WC program: reads input from the user and computes the number of characters, words, and lines in it.
- Proof of functional correctness of simple WC also provides **memory guarantees**.



- We can reason about **system calls** and other **non-deterministic** computations.
- We have libraries to reason about overlapping and disjoint memory regions. For example, we can verify whether the stack and heap overwrite the program or each other during execution or not.

We are working on commercial-size artifacts, but we are far away from complete verification of x86 implementations and large software systems like OS kernels.



# Re-Consideration of the Core Question

Are we working on a fool's errand?

- By 2025, could we mechanically verify an x86 implementation with “10%” of the performance of a commercial design?
- By 2025, could we mechanically verify an Linux-like operating system and applications with “10%” of the complexity of today's systems?

If the answer is **No**, then we will not have trustworthy components. We then need a **risk-mitigation strategy**.

If the answer is **Maybe**, then we should embark on a program to make trustworthy computing a reality. We need to remove IP and legal impediments to allow the free and **mathematically-accurate** exchange of hardware/software specifications and analyses.