

Memory Analysis and Machine-Code Verification Techniques for Multiprocessor Systems

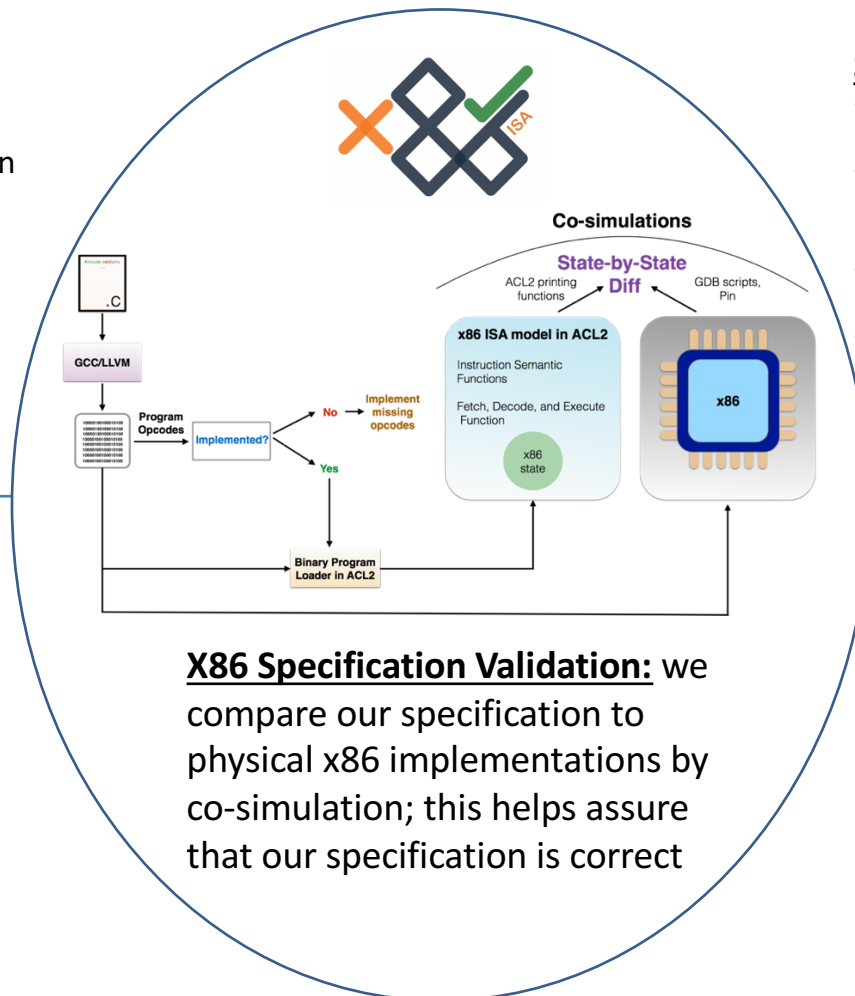


Challenge:

- Create a formal specification of the x86 instruction-set architecture
- Build formal verification system for x86 binary programs
- Provide x86 binary-level resource analysis system

Solution:

- Use ACL2 to write an executable, formal specification of x86 ISA
- Provide high-speed simulation while retaining clarity of specification
- Provide the means for formally verifying program behavior, memory usage, and execution time



Scientific Impact:

- Formal analysis and verification of x86 binary programs
- Provide memory-usage analysis tool (i.e., what memory addresses are used by program during runtime)
- Provide foundation to prove the correctness of object-file linking
- Formalize contract between programmers and implementers

Broader Impact:

- Improving x86 code quality possibly effects everyone that uses Windows, MacOS, and many Linux systems
- Our specification is already being used by industry
- Our x86 specification has been made public, and is distributed with the ACL2 system
- This is a build-to specification for hardware vendors and a compile-to specification for all compilers and systems that target x86 systems