Introduction to Formal Verification

P.Camurati G.Cabodi S.Nocco S.Quer

Formal Methods Group
Department of Computer Engineering
Politecnico di Torino
Torino, Italy

Outline

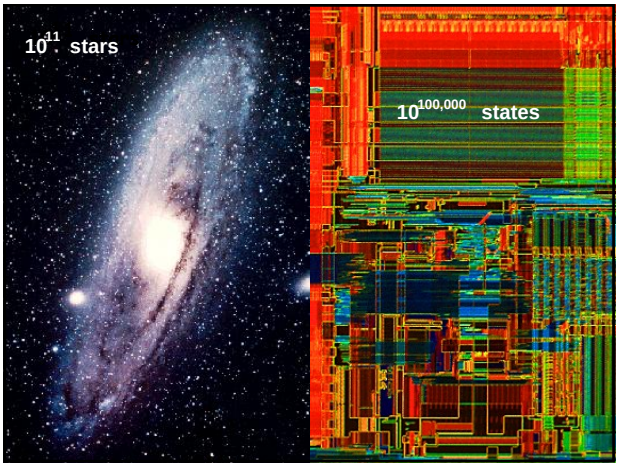
- Motivations
- Formal Verification

Outline

- Motivations
- Formal Verification

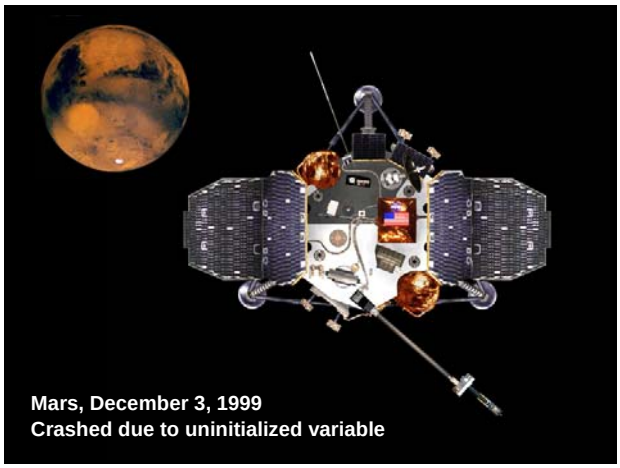
Motivations

- Digital systems continuously grow in scale and functionality
  - 10Mgates and beyond
  - Performance of integrated circuits (IC) doubling every year
  - Microprocessors containing 5M gates, doubling of frequency per generation, transistor scale by 30% per generation
  - Telecommunication chips are deep submicron application-specific integrated circuits (ASICs) with more than 1M gates
  - I/O pins limit observability and controllability, likelihood of design errors increasing



1994, Intel Pentium and Pentium Pro Microprocessors

- Problem with the floating point unit
- Cost of correction about $250 M
- In 1995, problem with TI 320C32 floating point digital signal processor



Outline

- Motivations
- Formal Verification

Formal Verification

- *Formal Methods*
  - mathematically-based languages, techniques, and tools for specifying and verifying systems
- Complement to simulation to improve design quality
- Increase understanding of a system by revealing *inconsistencies*, *ambiguities*, and *incompleteness* ... often even by just going through the process of rigorous specification

Terminology

- Formal Methods is the application of logic to the development of "correct" systems
- Correctness is classically viewed as two separate problems, validation and verification
  - Validation: answers "are we building the right system?"
  - Verification: answers "are we building the system right?"
- Formal Validation
  - Can we use logic to help ensuring that the specification is complete, consistent, and accurately captures the customer's requirements
- Formal Verification
  - Can we use logic to help ensuring that the system built faithfully implements its specification

Verification is an Industry-wide issue

Intel: Processor project verification:
"Billions of generated vectors"
"Our VHDL regression tests take 27 days to run. "

Sun: Sparc project verification:
Test suite ~1500 tests > 1 billion random simulation cycles
"A server ranch ~1200 SPARC CPUs"

Bull: Simulation including PwrPC 604
"Our simulations run at between 1-20 CPS."
"We need 100-1000 cps."

Cyrix : An x86 related project
"We need 50x Chronologic performance today."
"170 CPUs running simulations continuously"

Kodak: "hundreds of 3-4 hour RTL functional simulations"
Xerox: "Simulation runtime occupies ~3 weeks of a design cycle"
Ross: 125 Million Vector Regression tests

Design Teams are *Desperate* for Faster Simulation

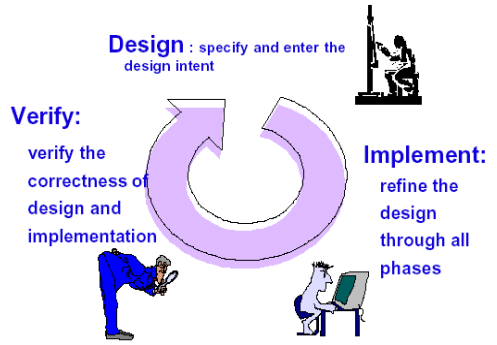
Application of Formal Verification

- Formal methods are used today in many applications including
  - Microprocessor Design
  - Cache Coherency Protocols
  - Telecommunications Protocols
  - Rail and Track Signaling
  - Security Protocols
  - Automotive Companies
  - ...

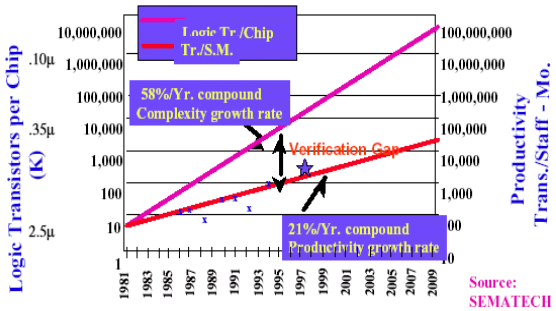
Verification

- Design Verification
- Implementation Verification
- Manufacture Verification (Test)

Design Verification



The Verification Gap (1/2)



Kurt Keutzer