White Paper on High Confidence Technology *

Shiu-Kai Chin
Dept. of Electrical & Computer Engineering
Syracuse University, Syracuse, New York 13244

Rome Laboratory - Computer Systems Branch
Griffiss Air Force Base, New York 13441

Abstract

One of the principal challenges to creating high-confidence systems, be they global-scale information infrastructures or high performance/scalable systems, is they must operate in a known and trustworthy manner. This requires the correct and timely integration of specifications and designs at all levels of detail. The needs of high-confidence systems have led to the development of mathematical tools sometimes known as "formal/high-assurance design methods." These tools have broad application to hardware, software, and system design. The successful development and deployment of high-confidence design methods could have as much impact on system design as the development of very-large- scale-integrated (VLSI) circuit technology had on computer design. The purpose of this white paper is to advocate for research programs in high-confidence design methods which lead to their industrialization and incorporation into standard engineering practice.

1 Introduction

System disasters like the Denver Airport baggage handling system, the flawed Intel Pentium processor, and the aborted FAA air traffic control system, speak to the difficulties of successfully fielding large systems. Large-scale system design is daunting because it requires the integration of many varied and heterogeneous parts into a single functioning entity.

The sheer number of details makes it impossible for any one person to fully comprehend such large systems in their final form. At the highest levels, even the properties of combined specifications are unknown. Unanticipated sub-system interactions introduced at the specification level remain undetected until late in the system-integration-and-test phase leading to costly delays, overruns, re-designs, and loss of life. Safety and security concerns like those related to aircraft flight controllers, medical electronics, secure networks, and information warfare, add more design complexity further stressing already inadequate design technologies.

High-confidence design technologies are needed which enable system designers and implementers to describe, implement, and verify systems at all levels. These technologies must capture behavioral descriptions, system properties, and design descriptions in ways which enable the creation and verification of implementations in a timely fashion. Above all, they must be practical, i.e. enable designers to create correct designs of higher quality more quickly when compared with today's methods.

Hints of high-confidence design technology are visible today in methods based on mathematical logic. Designs today are based on simple propositional or switching logic, i.e. on-or-off, true-or-false. Designs based on propositional logic can only be tested fully for safety and other properties by explicitly trying all possible cases - something which is infeasible for even modestly sized systems.

*Accepted for presentation at the Forum on Federal Information and Communications R & D, July 6–7, 1995, Bethesda, MD

More powerful logics like predicate calculus and higher-order logic have additional descriptive power due to their ability to make generalizations based on the internal structure of terms and parameterization. The added power of higher-order logic compared with simple switching logic is similar to the added power of algebra over simple arithmetic. This added power enables different descriptions, e.g. implementations and specifications, to be equated without having to test all possible cases explicitly. It also enables checking for safety and security properties over all modes of operation.

While more powerful logics are more complex, automated tools like model checkers and theorem-provers exist which provide computer-assisted reasoning, (CAR). CAR makes high-confidence design feasible in the same way computer-aided design (CAD) tools make VLSI design feasible. Both manage clerical details accurately and support the suppression of design details via abstraction to higher levels of description.

VLSI design technology successfully integrates two levels of design: integrated circuit design at the transistor level and logic design at the gate level. The key innovation is the means to create custom integrated circuits containing millions of transistors from gate-level logic designs described by switching logic formulas. The success of VLSI design is the foundation of the information infrastructure we have today.

High-confidence design technology based on mathematical logic has the potential of integrating all levels of design - from the gate level through networks of computers. Just as VLSI design enables the creation of custom chips containing millions of transistors, high- confidence design enables the creation of custom mathematical theories containing millions of mathematical theorems describing entire systems. Such technology enables the rapid and reliable development of complex systems. It provides accurate, reusable, and verifiable design descriptions at all levels. It assures that what the user expects is faithfully executed.

2 Problem Statement

While high-confidence design based on mathematical logic promises much, many challenges still remain.

2.1 Theoretical Challenges

Computer engineering has yet to identify what constitutes the "atom" in software design. The difference between software practice and mature systems engineering is the same difference between alchemy and chemistry. Alchemists lacking atomic theory could not predict the properties of the substances they combined. Chemists with atomic theory can.

Similarly, electrical engineers using the atomic notion of sinusoidal signals can predict the properties of subsystems assembled in some structure using Fourier analysis which is the foundation of linear systems theory. Because of the elegant composition properties of linear systems theory, electrical engineers go to great lengths to build components which respect the underlying superposition assumptions required by the theory.

Part of the problem today is that we design at the level of implementations, e.g. at the level of writing code. This is equivalent to designing a power plant by hooking up the turbines and transformers to see if the design works.

2.2 Practical and Educational Challenges

The principal practical problem facing high-confidence design methods is making the transition from mathematical theory to engineering practice. A common perception is these methods have been invented by mathematicians for mathematicians and are inaccessible to engineers. This perception is largely accurate. Engineers and mathematicians are separate communities. There is a lack of researchers who are knowledgeable in both engineering design practice and mathematical logic.

It is a misconception that engineers do not use formal methods. Schematics are formal objects. They have a logical interpretation and are related to logical formulae. Logic minimization algorithms are value-preserving transformations at the propositional (gate) level. The point is propositional logic has been refined and specialized to fit the design needs of engineers. A similar refinement and specialization is necessary and feasible for predicate calculus, higher-order logic, and language semantics.

The basis for such a refinement already exists. Over 15 years ago, Siewiorek, Bell, and Newell identified three main levels of abstraction. Each level is characterized by: 1) a distinct representation language; 2) components, modes of combination, and laws of behavior; and 3) a graphical representation coupled with a symbolic or algebraic representation. The levels, structures, and components are:

- Processor Memory Switch Level:
  - Structures: Networks of computers
  - Components: Processors, memories, and switches
- Program Level (High-Level Languages):
  - Structures: Programs
  - Components: Subroutines
- Program Level (Instruction-set processors):
  - Structures: Instruction sets
  - Components: Memory state, processor state, instruction decoders, instruction execution units.
- Logic Design Level
  - Structures: Processors
  - Components: sequencers, ALUs, registers, gates

While corporations like Intel, Microsoft, Motorola, and National Semiconductor are attempting to incorporate some high-confidence mathematical methods into their design processes, competitive pressures preclude them from making substantial investments of resources to fully learn and exploit their potential. This is similar to the early days of VLSI design when one heard comments like, "there are fewer VLSI designers than there are NFL football players." VLSI technology was transferred to US companies when student engineers were taught VLSI design as part of their training.

Today, there are fewer engineers skilled in formal/high-confidence design methods than there are NFL football players. What is needed are engineers who are trained to use these methods.

2.3 Programmatic Challenges

Current research sponsored in high-confidence computing and mathematical logic applied to design is isolated and fragmented. At Rome Laboratory, high-confidence technology is being funded by three separate groups under separate programs. These groups are the computer security group, the knowledge-based software engineering group, and the electronic reliability group. While each has valid objectives, the fragmentation detracts from achieving a critical mass necessary to solve the larger systems engineering problem.

3 Recommendations

3.1 Focus on Refining Theory to Practice

Research in high-confidence computing has largely been done by computer scientists and mathematicians. Reduction to practice requires the deliberate introduction of engineers, both academic and industrial, into this research area. Incentives can be made to encourage practical demonstrations by universities and industry which will flesh out the necessary details to incorporate high-confidence techniques into design practice. Examples of this in the VLSI area were the creation of the MOSIS fabrication service, the funding of CAD tool development, and the funding which incorporated VLSI design into engineering curricula.

The development of curricula, course material, and CAD tools are crucial. The book Introduction to VLSI Systems by Cal. Tech.'s Carver Mead and DARPA's Lynn Conway, was instrumental in transitioning VLSI technology into universities and industry. VLSI CAD tools like the Berkeley MAGIC CAD suite made the practice of VLSI practical. These tools were freely available to universities at little or no cost.

In the area of high-assurance design tools, several theorem-provers, model checkers, and specification refinement tools have been developed by DoD. Like the Berkeley CAD tools, they make the training of engineers in high-confidence design methods practical. Some of the tools, languages, and their associated design levels are listed below.

- Processor Memory Switch Level:
  - Languages: Calculus of Communicating Systems (CCS), Process Specification
- Language (PSL), higher-order logic.
  - Tools: Concurrency Workbench (CWB), Romulus, HOL.
- Program Level:
  - Languages: Larch, higher-order logic, ADA, C
  - Tools: Penelope, HOL
- Logic Level:
  - Languages: Larch, higher-order logic, VHDL
  - Tools: Penelope, HOL

Continued development of these tools is necessary emphasizing integration across design levels and integration with existing system design processes and tools.

These tools should be made available to universities. Rome Laboratory tools have been used successfully at Syracuse University and other institutions to teach graduate computer engineers high-confidence design methods.

Currently, Syracuse University engineering students are working with Rome Laboratory engineers on a 32-bit processor designed and verified using formal techniques based on higher-order logic. This processor should be fabricated at MOSIS by the end of this year. This effort has shown that high-assurance design techniques produce designs of higher quality in less time when compared with conventional techniques.

3.2 Focus on Finding a "Linear Systems Theory" for Computing

This is sometimes known as the "new mathematics" for computing. What is striking when one examines the sister discipline of electrical engineering is that it has a block diagram algebra while computer engineering does not. The block diagram algebra in electrical engineering is built upon Laplace and Fourier transforms which are based on the underlying atomic concept of combining sinusoidal signals. It is applied to power systems, communications, signal processing, and controls. The key attribute of linear systems theory is its compositional semantics which allows structural descriptions to be equated to purely behavioral (system transfer function) descriptions. Research to produce a similar theory for computer engineering is necessary.

While it is too early to do more than speculate, the notion of "instruction" may be a candidate atomic notion. Instructions are not programming language commands. They are the commands which affect the state and outputs of objects. These objects are microprocessors, C++ objects, or FORTRAN subroutines. Instruction-set behavior defines how memory and outputs are affected by instructions.

ACM Turing Award winners like Hoare and Milner have settled on "events" or "instructions" as atomic elements in concurrent processes. They have developed algebras to deduce the behavior of composed processes.

It could be that process algebras may be to software as linear systems theory is to electrical engineering. Current "clean room" techniques at the code level may correspond to detailed circuit analysis.

Another theoretical problem is the combination of various logical systems to support a continuous design process across all levels of design abstraction. Current VLSI CAD tools span the levels from logic to layout which effectively integrate propositional calculus, transistor models at the switch level, and differential equations at the circuit level. The point is that there is not a common language, representation, or single vendor CAD environment. A similar integration of heterogeneous logical systems should be anticipated for system design. How consistency is maintained across the interfaces must be solved.

3.3 Unify Research Programs Related to High-Confidence Computing

Research in high-confidence design technologies are spread over programs in computer security, software engineering, reliability, and fault-tolerance. The danger of this approach is that research cuts in one area affects progress in other areas. For example, research on a theorem-prover associated with Larch was cut by NSA. This directly affected an effort on a Rome Laboratory Small Business Innovative Research contract which linked Larch to VHDL. The inter-related nature of the efforts was not known as part of the budget planning.

There is little redundancy of research efforts in this area. Cuts in one area compromise efforts in others. Focused and coordinated attention is required to make high-confidence design methods a mature technology suitable for technology transfer.

4 Conclusions

High-confidence design methods are on the verge of practicality. This technology will have as much impact on system design as very large scale integrated circuit design had on computer design. Successful deployment of high-confidence design methods will bring significant savings in time and cost associated with fielding large and complex systems in addition to assuring the systems have the necessary safety and security properties. A focused effort on refining computer science theory to engineering practice is needed. The pieces for such an effort exists, these pieces must be integrated to realize the full benefits of this technology.