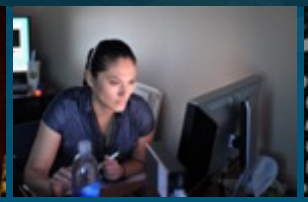
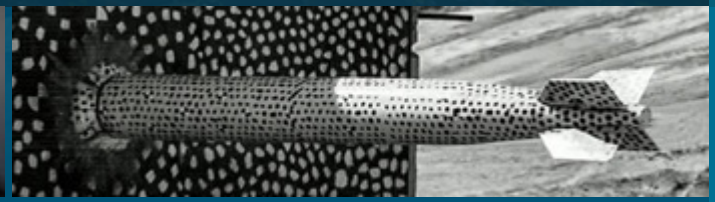
UUR

Sandia National Laboratories

Formal Methods in PSAAP IV: Improving Assurance of Cyber-Physical Systems





PRESENTED BY

Robert Armstrong

Controlled by:

U.S. DEPARTMENT OF ENERGY | NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

NUCLEAR DETERRENCE

Responsibilities form a critical mandate

Warhead systems engineering & integration

- Systems modeling, analysis

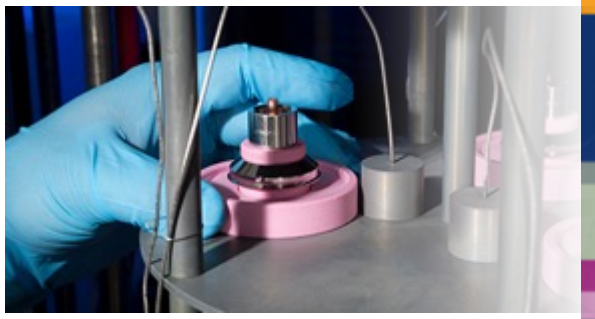
Design agency for nonnuclear components

- Radar
- Safety systems
- Arming, fuzing, and firing systems
- Neutron generators

Multidisciplinary capabilities

Required for design, qualification, production, surveillance, computation/ experimentation

- Major environmental test facilities & diagnostics
- Materials sciences
- Light-initiated high explosives
- Computational analytics



Production agency

- Neutron generators
- Microelectronics
- Thermal batteries

Tri-lab Mission Assurance organization proactively prevents defects and ensures mission success

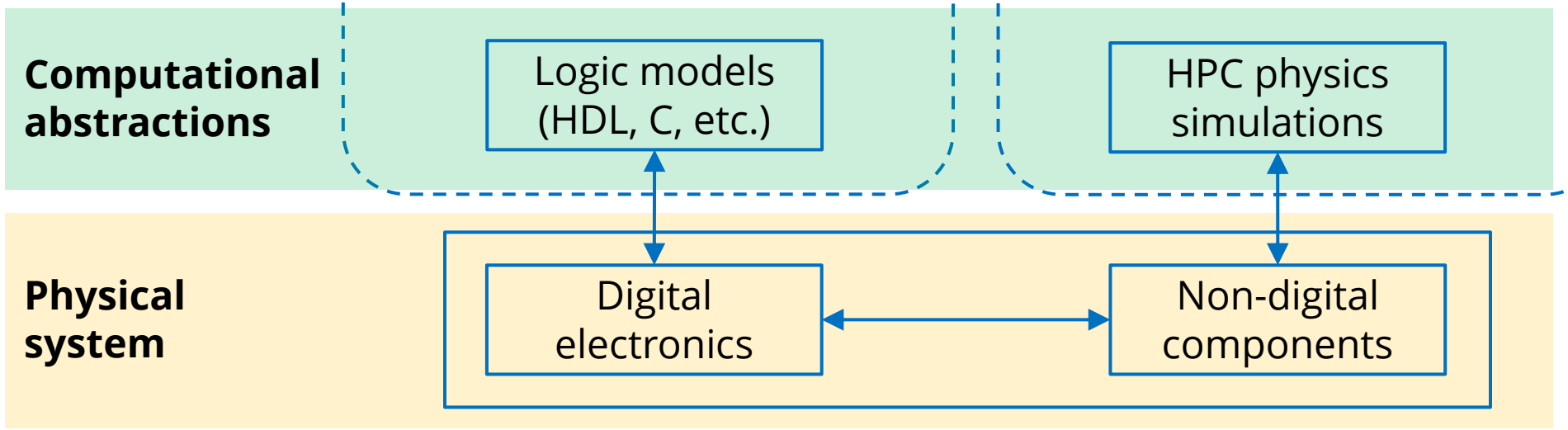
Role of Digital Computation in HPC & High-Consequence Control Systems

Embedded computation
(new in PSAAP IV)

- Directly instantiated in the system – correctness is critical
- Relatively simple **control logic**
- Key question: Does the digital behavior meet the system requirements?
- Main challenge: Ensuring **strict**, comprehensive correctness

Simulation
(traditional PSAAP)

- Used to guide design – not directly in the system
- **Numerical** modeling of physics (traditional V&V applies)
- Key question: Does simulation accurately represent the physics?
- Main challenge: **Scale** and complexity of software



In both cases, **formal methods** can eliminate bugs that **testing alone** cannot!

Formal Methods enables **co-design of exascale HPC** special-purpose hardware and software

National Security Enterprise (NSE) concerns involve not only digital-systems, but also complex cyber-physical systems that all labs share

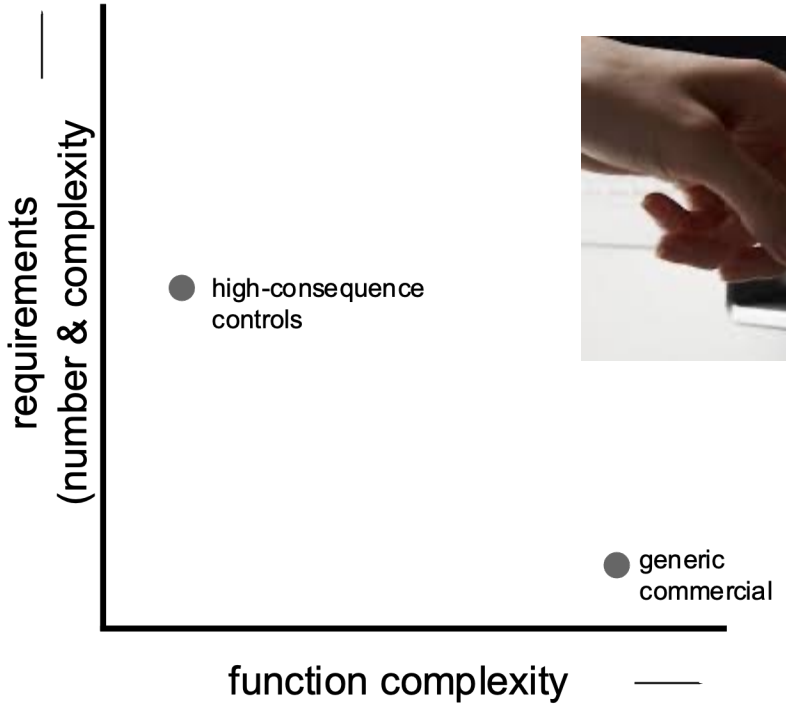
- NSE is responsible for the highest-consequence systems in the world
  - The digital components implementor and systems integrator
  - Responsible for end-to-end correctness

- Digital systems ***are*** complex systems
  - most of the problems we are trying to solve are due to complexity

- Systems engineering occurs at all DOE labs/sites
  - High-consequence systems are Complex Systems due to high-reliability requirements
  - Like digital systems, NW Systems inherit their worst problems from Complexity
    - Cascading failures
    - Corner cases and unexpected vulnerabilities
    - All our digital systems are cyber-physical systems anyway
  - Formal tools used to reason about digital systems can also be used in Systems Engineering
  - Ensure correctness of designs of full Systems and implementations
    - Operational Tech
    - Supply chain

Requirements for Systems of Interest

- Our control systems are mostly low complexity, relatively easy to analyze, like a dishwasher.
- But, they often have a large number of complex, high-consequence safety, security, and reliability requirements.
- Low complexity + high consequence + complex requirements = ideal for a high-assurance formal approach to design and/or verification.



Heavily Computationally Constrained: Back to the ’80s Future

- A typical system is simple, dumb, and resource constrained
- We build from scratch
- Our own fab, our own processor, our own peripherals
- Processor:
  - 5-10 MHz (can go 10-50 MHz, for higher requirements, or kHz for low power)
  - ~Mbytes of RAM
  - ~100kbytes total storage for boot images
  - No MMU
- We write custom firmware to drive this currently
- Assessment for surveillance activities of existing systems
- Analysis, verification and qualification of new systems

GOAL: Digital Assurance of High-Consequence Digital Systems with One QED (Top to bottom proof of correctness)

High-consequence systems with digital components must be shown to be correct with respect to their requirements

Testing and simulation alone cannot explore the entire state space or provide evidence of correctness

Designer's creativity …

is expressed in the system specification …

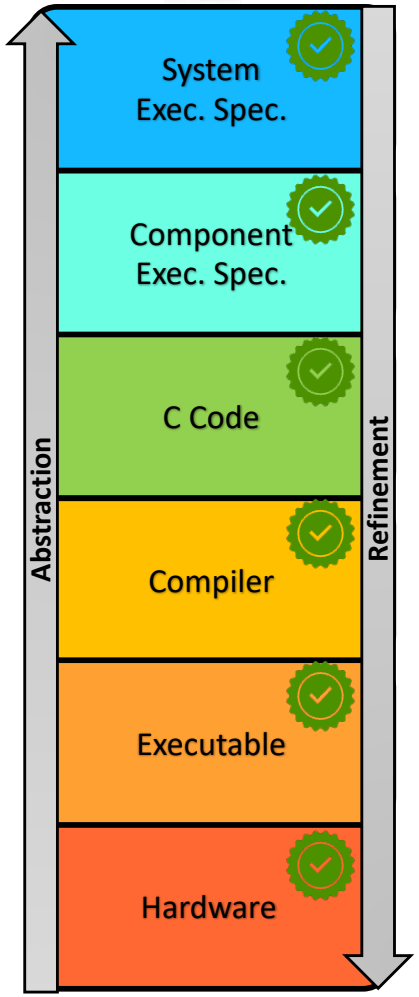
… is expressed in the component specification …

… behavior capture in source code …

… is certified to be the same …

… as it is here

… as it is here



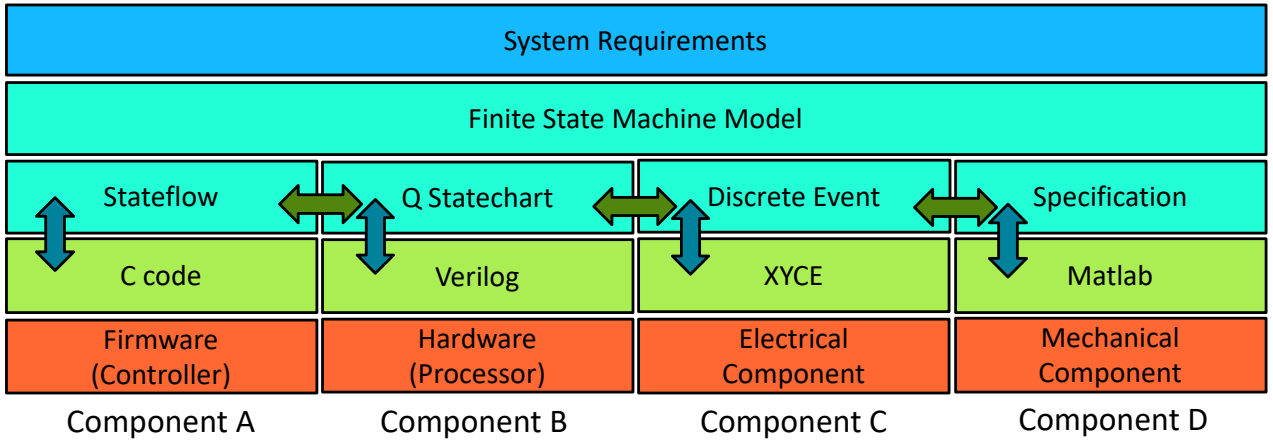
To ensure correctness:

- Each abstract representation of the digital component must be proven correct
- A rigorous mathematical analysis must provide evidence of correspondence between different abstraction levels

Correctness is a way to eliminate defects and vulnerabilities that can be exploited by our adversaries

Formally Informed Model-Based Systems Engineering (MBSE) Leverages Multiple Levels of Modeling

- Represent systems and components at appropriate levels of abstraction to optimize tractability and fidelity of analysis **throughout** the design process
  - From highly abstracted, e.g., discrete-event model of a system-level specification
  - To highly detailed, e.g., logic gates

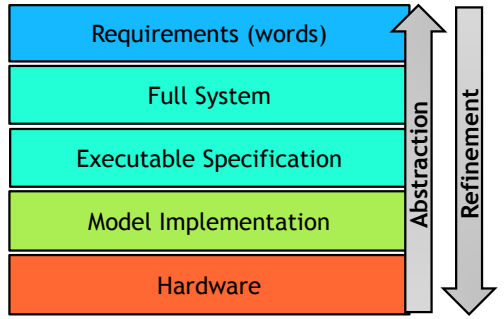


- Traditional modeling simulation approaches are **bottom-up**
  - Confirmatory analysis of a largely complete design

Correspondence achieved via principal abstraction or formal verification of executable specification and model implementation

Interface specification and verification must correspond to full system finite state machine model

- MBSE adds **top-down** and allows combining both directions iteratively
  - Apply modeling *before* the detailed design is determined – provide guidance early and often
  - Explore high-level **design choices** quantitatively by using system models as virtual prototypes

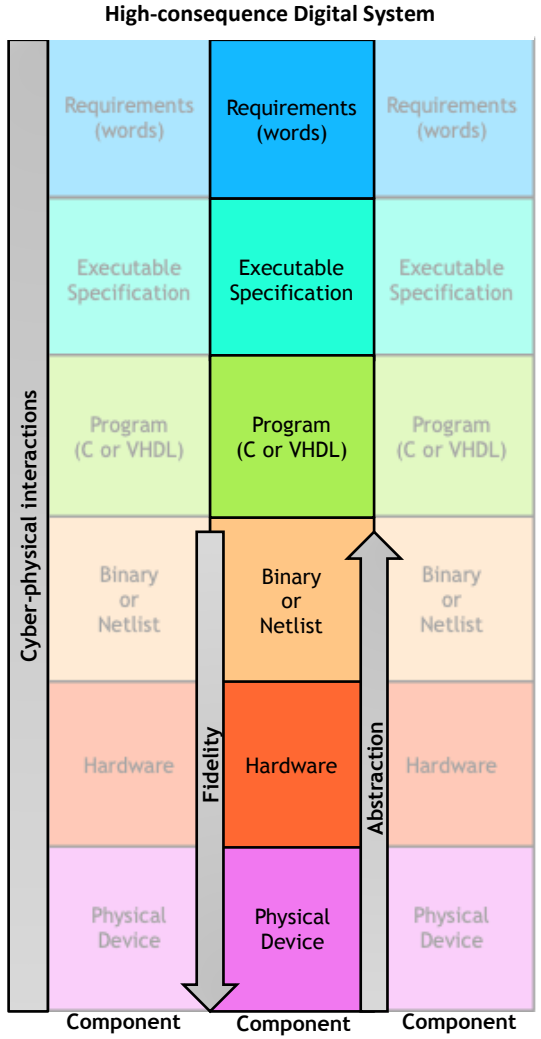


PSAAP IV **Formal Methods**
Research and Capability Development Needs

Goal: Building on ASC formal methods R&D to date, enable digital assurance for current and future **high-consequence digital systems**.

Background: Digital design flaws could result in exploitable vulnerabilities. "Formal methods" allow implementing high-consequence digital components with **mathematically proven** reliability, safety, and security (not achievable by testing alone). This approach has already prevented defects and improved assurance of systems of interest.

Current and future needs: Enable comprehensive high-consequence digital system verification with increased **rigor**, improved **usability**, and broader **applicability** within the engineering community.



Digital systems design process R&D needs

- Extend **Model-Based Systems Engineering** tools to enable formally assured design
- Establish digital workflow to derive and verify rigorous **cybersecurity** requirements
- Enable richer **"executable specifications"** relating system and component behavior precisely
- Enable scalable analysis of future **modular designs** such as Distributed Bus-Based Architectures
- Assure future embedded **software** by verifying algorithms and compilers
- Assure future embedded **hardware** by verifying processors and synthesis tools
- Analyze **"cyber-physical"** behavior where digital logic meets analog/continuous physics

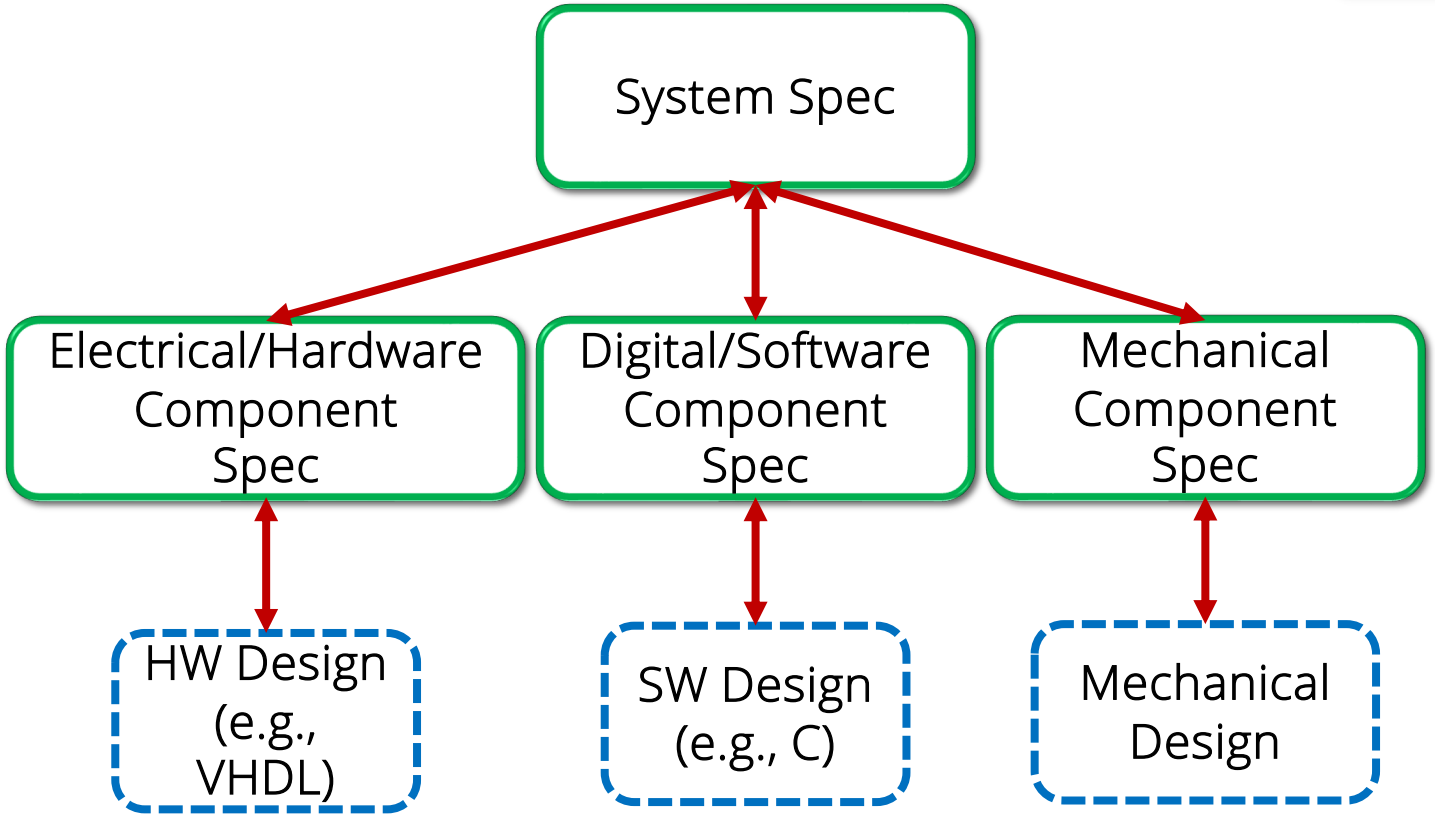
Research Areas of Interest

- **Foundational reasoning** about realistic **reactive systems**: how do we articulate and prove properties about how combined software/hardware systems interact with other systems and the world
- Reducing **trusted computing** base for HPC accelerator and embedded systems
- **Soundly** incorporating uncertainty (e.g. from sensors or HPC simulations) into verified systems
- **Verifying cyber-physical systems** where both physical and digital parts are non trivial
- Developed **formal model-based design tools** targeted for engineers, for **constructions and analysis** of digital systems specification
- Add **rigor** to system and component specifications to provide early assurance that the system will meet its requirements
- **Improve compositionality** of different components specifications, properties and proofs of correctness
- Mathematically rigorous and **formally verifiable characterization** of cyber security properties for embedded systems
- Formally verified functional requirement of embedded **software/hardware** and their **interfaces** via static analysis and model checking
- **Improve automation** of software formal verification both for source and binary code
- Develop theory and tools for formal reasoning for **floating point, data flow, information leakage**
- Address **scalability limitations** in SMT/SAT solvers, symbolic execution tools and refinement proofs, as well as the generation of corresponding **proof certificates**
- Develop **quantum resilient** implementation of **cryptography operations** with corresponding formal specifications, and proofs of correctness
- **ML** applied to **optimization** on certified compilers, automation of **proof search/repair**, and generation formal model for analysis of **high-consequence systems** behaviors.

Design Specifications Constrain Allowed Behaviors

Component co-design must not violate constraints specified at system level

Correctness of design with respect to requirements should be demonstrated at the highest possible level of abstraction (lowest complexity)



~~Document~~
~~Descriptive Model~~
Executable Specification

~~Traceability~~
Mathematical Refinement

Engineering Implementation


QUESTIONS