SAND2016-2079PE

Exceptional service in the national interest

Sandia National Laboratories

Formal Verification of Digial ASICs

Jason Michnovicz

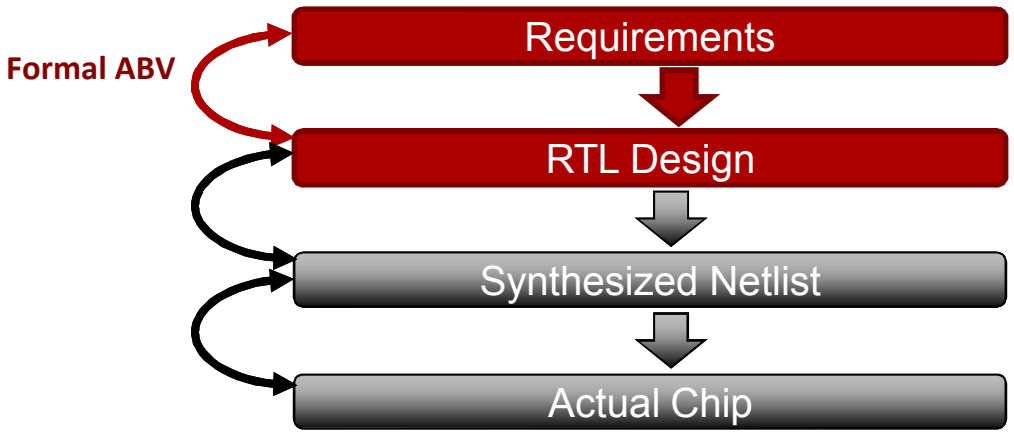
U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Formal ASIC Verification at Sandia

Sandia National Laboratories



- **We use Formal Assertion-Based Verification**
  - **Tools: RTL model-checking tool with SystemVerilog Assertions**
    - **JasperGold Formal Property Verification App**
    - **OneSpin DV-Verify**
  - **Scope: ASIC requirements (system-level properties)**
    - **Of special interest – properties about safety of high-consequence designs**
  - **Formal Verification is a key piece of the ASIC verification strategy for some of our designs, but not the only technique we use**

Current Value Added

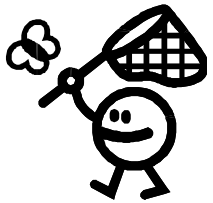
Sandia National Laboratories

- **Formal verification is a success at Sandia**
  - **We have successfully proved critical requirements**
  - **We have increased confidence in design due to extra scrutiny**
  - **We have impacted designs**
    - **Found some corner cases which resulted in bug fixes after traditional verification was completed**
    - **Helped argue to (slightly) clarify some requirement wording**

Obstacles

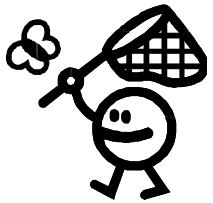
Sandia National Laboratories

- **... But our current methods are unsatisfying**
  - **Requirements are not written for verifiability**
    - **Too imprecise to express formally**
    - **Too high-level to assert directly**
    - **Described in terms of constructs that don't exist at that scope**
    - **Incomplete**
  - **Tools are not designed to prove system-level properties**
  - **Our workaround process leaves a lot of room for human error**
    - **Sometimes we can't prove the requirement as stated, so we have to prove the designer's intent**
    - **Need to manually break down each system-level requirement into a large set of register-level properties**
      - **Must informally argue that the low-level properties imply the top-level property**
      - **Introduces traceability concerns**
      - **Documentation becomes an unwieldy "proof by intimidation"**



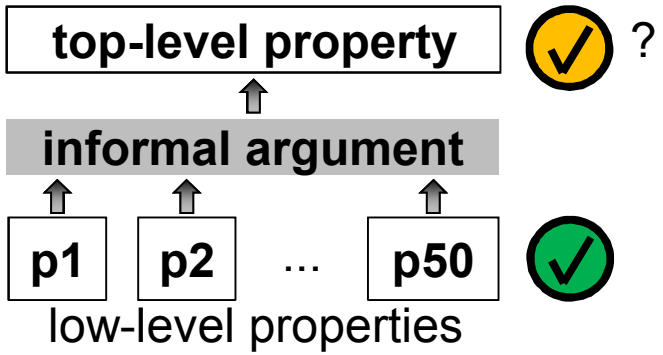
Obstacles

- **... But our current methods are unsatisfying**
  - **Requirements are not written for verifiability**
    - **Too imprecise to express formally**
    - **Too high-level to assert directly**
    - **Described in terms of constructs that don't exist at that scope**
    - **Incomplete**
  - **Tools are not designed to prove system-level properties**
  - **Our workaround process leaves a lot of room for human error**
    - **Can't technically prove the requirement, instead prove the designer's intent**
    - **Need to manually break down into a large set of register-level properties**



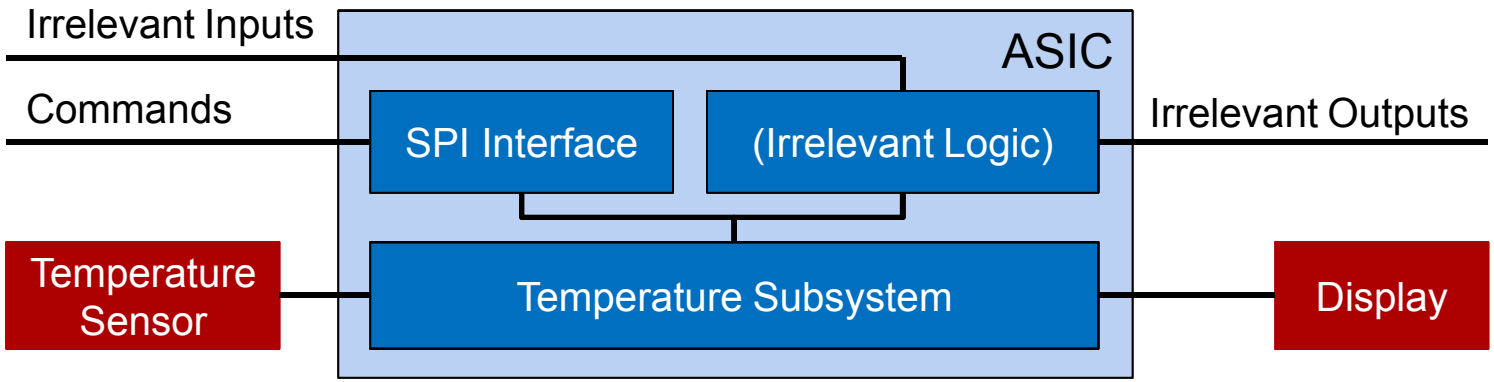
What we want

What we have



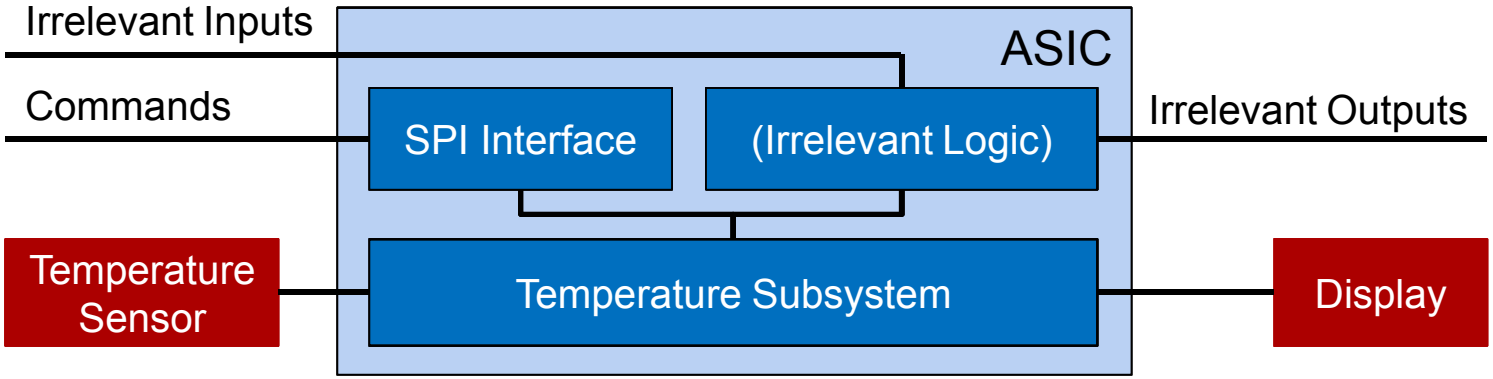
Example

- **Consider an ASIC which:**
  - **Receives data from several sensors, including a temperature sensor**
  - **Receives commands over a SPI bus**
  - **Continually displays the temperature until it recieves the "stop temperature display" command**
  - **Freezes display once "stop temperature display" command is received**
- **Target requirement**
  - ***All SPI commands shall not affect the temperature subsystem except the "stop temperature display" command***



Example

- **Ambiguity**
  - **“Shall not affect” – this can’t be expressed in SVA!**
  - **“Temperature subsystem” – this assumes a specific internal architecture!**
- **Approach**
  - ***Prove that if the “stop temperature display” command is not received, then the output to the display is a function only of the temperature sensor inputs***
  - **The function above is calculated by a nontrivial algorithm, so it has to be proven in terms of lower level sequential logic steps**
  - **The “stop temperature display” command has to be precisely defined as a specific family of sequences of activity on the command wires**



Potential Improvements

Sandia National Laboratories

- **What might help? (speculative)**
  - **Requirements written more precisely**
  - **Designs created by refinement**
    - **Requirements hold by construction**
  - **More support for higher level abstractions in the RTL formal verification tools and languages**
  - **A more sophisticated proof environment**
    - **Have a framework for compositional reasoning instead of relying on an informal argument**