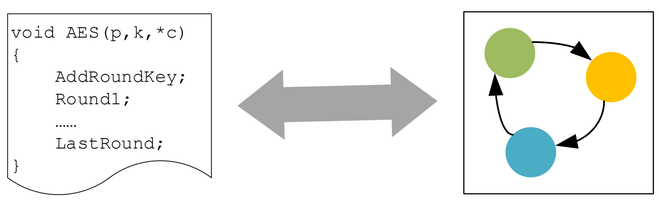
Equivalence Checking Using the Intermediate Instruction-Level Abstraction

Hongce Zhang, Sharad Malik
Princeton University

Motivation

- **Top-down design methodology**
  - First, system level description
  - Next, register-transfer level (RTL) description
  - Then, gate-level, ……
- **Verification of low level designs is much more difficult**
  - Start from a correct high-level design
  - Ensure equivalence in each following step



Intuition

- **Hardware's perspective**
  - State update in response to the input
- **Software's perspective**
  - Executing instructions
- **Extract an instruction-level abstraction from the hardware, and translate software description also into ILA**
  - Compare at the same level
- **Hardware implementation contains more details than needed**
  - Need a way for abstraction

Synthesize ILA

- **No need to build full abstraction from scratch**
- **Template-based synthesis**

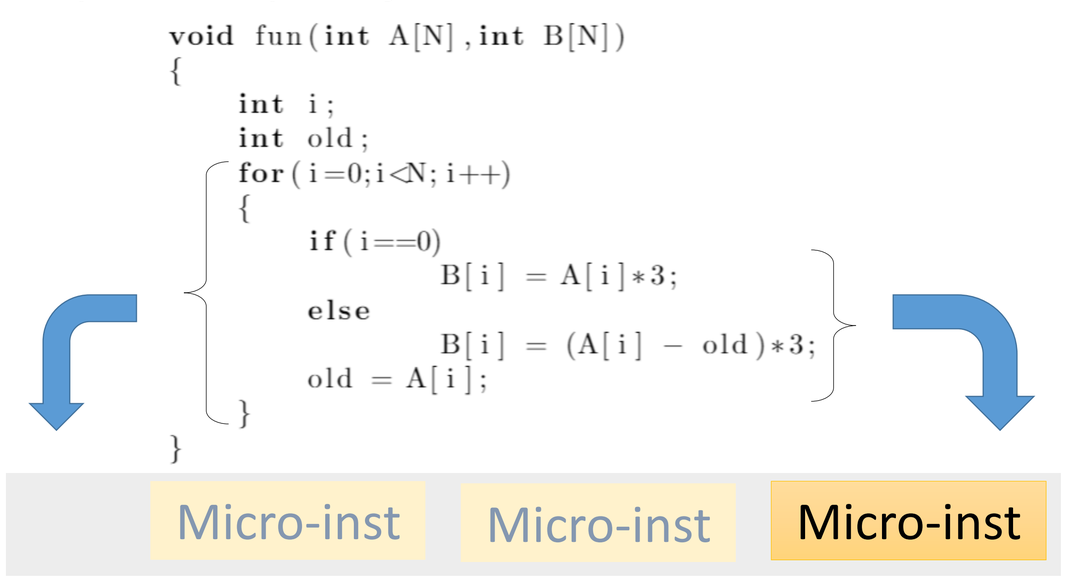
  Template:
  - Describing states (register / memory / …) that we care about
  - Transition function with "holes"

  Refinement relations:
  - Abstract away invisible / don't care state transition
  - Specify when and how ILA and RTL should match

  Simulator:
  - Answer queries
  - Automatically constructed by adding interface wrapper to a given RTL

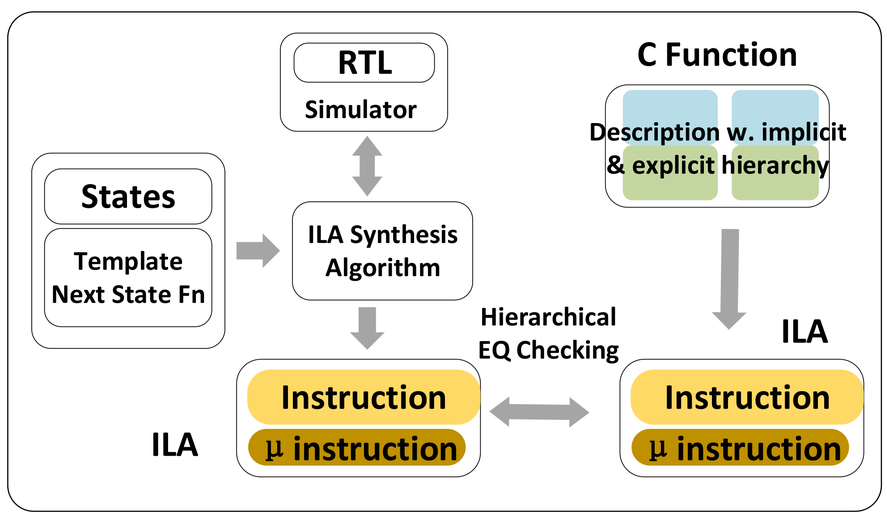
Hierarchical Equivalence

- **Exploit hierarchical structure to save efforts in equivalence checking**
  - Subroutine invocation
  - Loop and loop body



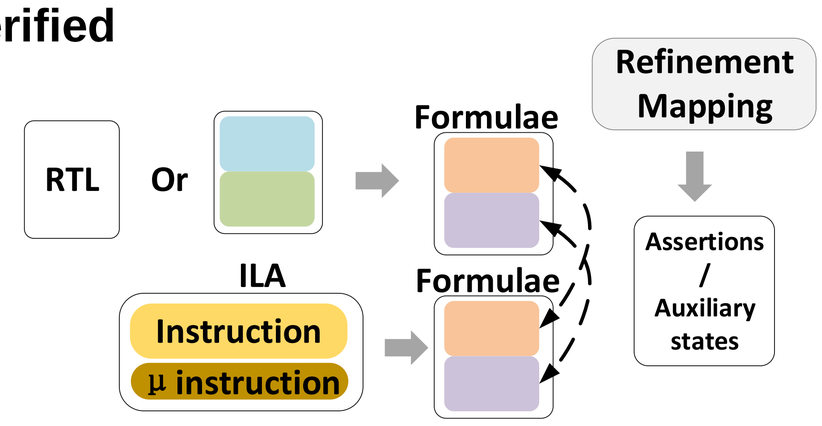
Flowchart

- **Extract ILA from RTL and translate C into ILA**
- **Find the lowest level of similarity and check bottom-up**



Discussion

- **Process of synthesis and translation is also verified**



- **Similarity of hierarchy helps save efforts in ILA vs ILA checking, but not necessary.**
- **Last resort: compare on the highest level**

PRINCETON UNIVERSITY

This work was supported in part by the Center for Future Architectures Research (C-FAR), one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.