**RTLCheck: Verifying the Memory Consistency of RTL Designs**

Yatin A. Manerkar, Daniel Lustig*, Margaret Martonosi, Michael Pellauer*

*NVIDIA

---

**Motivation**

How to ensure RTL maintains consistency orderings?

---

**From Microarchitecture to RTL**

- **Axiomatic Microarchitectural Verification**
  - Axiosim \(_\#B_FILO\):forall microops \(\#a\), \(\#b\):
  - \(\text{OnCore}(c, a) \land \text{OnCore}(c, b) \Rightarrow \text{SameMicroop}(a, b) \land \text{ProgramOrder}(a, b))\) \(\Rightarrow\)
  - AddEdge\((\{a, b\}, \{a, b\})\).

---

**Temporal** RTL Verification (using SVA, etc)

assert property \(!\text{loadBeforeWrite}\) (i) (if \(\text{Load}(i, \text{PC} DL) \Rightarrow 1\) then
  - \(\text{next cycle}(\text{Core}[1], \text{data} = 1)\).

---

**Lack of Outcome Filtering in SVA**

Simplified property (using outcome filtering):

BeforeAllWrites:

1. Load hasn't occurred yet!
2. St happens before Load!
3. Load does not read 0
4. St happens before Load!

Solution: Translate all cases, keep track of load value constraints (data restrictions for a given edge)

---

**RTLCheck Block Diagram**

Mapping Functions translatearch primitives to their RTL equivalents:

**RTLDesign**

- Node Mapping Function
- Litmus Test
- ispec Microarch. Assumps
- Program Mapping Function

---

RTLCheck

Temporal SystemVerilog Assertions (SVA)

JasperGold (RTL Verifyer)

Proven? Counterexample?

---

**Multi-V-scale: A Simple Multicore**

3-stage in-order pipelines:

- Core 0
- Core 1
- Core 2
- Core 3

- IF
- DX
- WB
- IF
- DX
- WB
- IF
- DX
- WB

Arbiter

Memory

---

**V-scale Bug and Runtimes**

- RTLCheck discovered bug in V-scale’s memory implementation
- When two stores are sent to mem in successive cycles, first store is dropped
- Runtime to verification below (max. 11 hrs/test)
- Avg. runtime 6.2 hrs/test
- Up to 89% of properties fully proven

---

**Conclusions**

- RTL must be verified against formal specifications of MCM behaviour
- **RTLCheck**: Automated MCM verification of arbitrary RTL against arbitrary microarchitectural orderings
- Translates microarch. axioms to equivalent temporal SVA properties
- Can handle arbitrary ISA-level MCMs (SC, TSO, ARM, …)
- JasperGold proves most of generated properties in minutes or hours
- RTLCheck enables full-stack HLL-to-RTL, MCM verification (with rest of Check suite) across a collection of litmus tests

---

**The Axiomatic-Temporal Mismatch**

- Axiomatic models check executions as a unit
- Axioms applied to an entire execution
- Easy and efficient outcome filtering
- Temporal models generate executions step-by-step as a “tree”
- Eliminating executions by outcome requires expensive global analysis
- SVA verifiers approximate by only checking constraints up to current step
- Makes outcome filtering impossible