CPUs are prone to memory reordering bugs which impact both correctness and performance, e.g.:
- AMD Phenom TLB bug
- Intel Haswell/Broadwell TSX bug
- Support ISTC-CC Goals:
  - Specialization: need to verify increasingly diverse architectural designs
  - Automation: PipeCheck verification is entirely automated in software
  - Big Data/to the Edge: code and/or data migration assumes memory consistency correctness to guarantee data arrival and readiness conditions

Unfortunately, testing cannot guarantee 100% accuracy depending on the model. Some proposed outcomes may be forbidden or permitted, depending on the model:

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Proposed Result in above litmus test is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC, TSO</td>
<td>Forbidden</td>
</tr>
<tr>
<td>ARM, Power</td>
<td>Permitted</td>
</tr>
</tbody>
</table>

PipeCheck can verify microarchitectural “happens-before” graphs against architectural expectations. Methodology and automated tool for verifying that a microarchitecture correctly implements the memory ordering rules of the specification.

Runtime per mu-proposal:
1. Methodology and automated tool for verifying that a microarchitecture correctly implements the memory ordering rules of the specification.
2. Treat preserved program order (PPO) as a proposition to be verified, rather than simply as an assumption.
3. Compare microarchitectural “happens-before” graphs against architectural expectations.

PipeCheck Overview

Rules in a memory consistency model:
1. Preserved Program Order (PPO): the set of orderings enforced by default
2. Fences: used to enforce orderings not enforced by PPO
3. Dependencies
4. Reading writes early, e.g., from a processor’s own store buffer

Litmus Tests

- Most common existing approach for verification
  - Example for Ld→Ld and St→St reordering:
  - Proposed outcome may be forbidden or permitted, depending on the model:

PipeCheck μ-Happens-Before Graphs

μhb = “microarchitecturally happens before”
- Cycle in μhb graph → the proposed execution is forbidden
- No cycle in μhb graph → the proposed execution is allowed
- Explains why a particular outcome is allowed or forbidden

PipeCheck microarchitectural analysis of the litmus test on the given pipeline

Verification Results

- Tested four real and simulated pipelines specified to be TSO
- PipeCheck verification is fast:

Litmus tests highlight discrepancies between arch. and parch. specs:

PipeCheck supports:
- inter-dependence of coherence and consistency
- techniques such as speculative load reordering, even when they technically violate the architectural-level specification
- Full details described in paper

Conclusion

We define microarchitecture-level happens-before graphs and use them to verify pipeline implementations against their architectural memory model specifications.

PipeCheck tool performs verification automatically given both specs